0 - 315 - 05927 - 3

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

• Please print or type — Écrire en lettres moulées ou dactylographier

Full Name of Author — Nom complet de l'auteur

STEPHEN SIU-KI AU

Date of Birth — Date de naissance

22/12/1954

Country of Birth — Lieu de naissance

HONG KONG

Permanent Address — Résidence fixe

75, JERVOIS STREET,
5TH FLOOR, CENTRAL DISTRICT,
HONG KONG

Title of Thesis — Titre de la thèse

AN INVESTIGATION OF THE PROPAGATION OF A FOCUSED LASER BEAM IN A MAGNETICALLY CONFINED PLASMA USING RAY TRACING TECHNIQUES

University — Université

UNIVERSITY OF ALBERTA

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

M. Sc.

Year this degree conferred — Année d'obtention de ce grade

1981

Name of Supervisor — Nom du directeur de thèse

Dr. C.R. JAMES and Dr. C.E. CAPJACK

Date

7/7/81

Signature

Au Siu Ki

NL-91 (4/77)

THE UNIVERSITY OF ALBERTA

An investigation of the propagation of a focused laser beam in a magnetically confined plasma using ray tracing techniques

by

© Siu-Ki Stephen Au

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF Master of Science

IN

Plasma Studies

Department of Electrical Engineering

EDMONTON, ALBERTA

Fall, 1981

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR Siu-Ki Stephen Au

TITLE OF THESIS An investigation of the propagation of a focused laser beam in a magnetically confined plasma using ray tracing techniques

DEGREE FOR WHICH THESIS WAS PRESENTED Master of Science

YEAR THIS DEGREE GRANTED Fall 1981

(SIGNED) Au Siu Ki

PERMANENT ADDRESS:

75, Jervois Street,

5th Floor, Central District,

Hong Kong.

DATED 24th June, 1981.

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled An investigation of the propagation of a focused laser beam in a magnetically confined plasma using ray tracing techniques submitted by Siu-Ki Stephen Au in partial fulfilment of the requirements for the degree of Master of Science in Plasma Studies

Supervisor

Supervisor

Date 24th June, 1981.

Dedication

To my parents

and

To the goodwill of mankind

Abstract

The propagation of a focused laser beam in a magnetically confined plasma is studied using ray tracing techniques. The transport of beam energy along rays in a vacuum is described in terms of a phase space distribution function which includes the beam diffraction effects. Solutions for ray trajectories in the plasma are solved for various density profiles. These solutions are used to match into corresponding density distributions within the medium. Energy deposition and ponderomotive forces are evaluated accordingly. Program packages are designed to be used with the magnetic shell flux code developed by McMullin, Milroy and Capjack.

Acknowledgement

It is a pleasure to express my gratitude to my supervisors: Dr. C.R. James and Dr C.E. Capjack for their intellectual and financial assistance. I also greatly appreciate their patience in giving me tremendous guidance. I am heavily indebted to Dr. J.N. McMullin for his invaluable ideas and discussions. Had he not given me the ideas, this thesis would not be accomplished.

Thanks are also addressed to Mr. A. Soliman for his generosity in spending tremendous amount of time to read through my work and for giving me encouragement.

Honors are also addressed to Mr. D. Way-Nee for his invaluable assistance in proofreading and in arranging necessary equipment for the preparation of this work. Special thanks are also directed to Dr. A.A. Offenberger and Dr. P.R. Smy for their discussions and consultations.

I would also like to thank the secretaries of the department, especially Ms B. Gallaiford and Mrs. S. Neuman for their assistance throughout this work. Finally, thanks are also due to my brother, Mr. Shiu-Kong L. Au, as well as Mr. T. Casey and Mr. T. Butler for there assistance in the preparation of this thesis.

Table of Contents

List of Figures

Figure Page

Chapter 1

Introduction

1.1 Significance of the problem

The utilization of lasers as energy sources for heating plasmas in a thermonuclear fusion reaction has been studied extensively in the past years. J. Nuckolls et. al[1] proposed the scheme of using laser beams to heat plasmas confined within a spherical pellet (inertial confinement), while Dawson[2] et. al. reported the feasibility of heating magnetically confined plasmas by lasers(laser heating with magnetic confinement). In both schemes, as the laser beam propagates within the plasma, energy is transferred to the medium. The plasma is heated up upon absorbing the energy and as a result, thermo-expansion of the plasma will take place. The disturbance in the plasma density affects the refractive index and hence a change in the propagation of the beam. This, in turn, will change the heating pattern for the plasma and a self-consistent system will be set up between the plasma and the beam. In order to understand the heating mechanism, it is essential to study the problem of laser beam propagation within a plasma medium. Before introducing the research objectives, a survey of the recent investigations on this problem will be carried out in the following section.

1.2 Review of past work

In 1974, Humphries[3] studied the propagation of a laser beam parallel to the magnetic field in a θ pinch plasma column. He found that waveguide type solutions are necessary for describing the beam condition within a plasma which was assumed to have a radially parabolic density profile. Mani et.al[4] used the method of normal mode analysis to describe laser beam propagation within a plasma column having a parabolic density variation. They evaluated the maximum number of modes that can be trapped within the plasma. Their analysis showed that the beam propagates periodically along the column. Feit et.al[5,6,7,8] developed a self-consistent treatment for the relation between laser beam propagation and plasma hydrodynamics. They found that the beam trapped itself as it heats up the medium. In their further investigations, they showed that an axial density changes were also found in the beam after it traversed the column. They also studied the case of a beam propagating through a region with an electron density profile determined from a detailed time-dependent hydrodynamic calculation. The beam was found to exhibit aperiodic properties along the column but remained trapped. McMullin et.al.[9] used normal mode analysis to derive a general expression for the field of a laser beam in a quadratic waveguide. The electric field amplitude was shown to exhibit periodic property along the column. For an axially varying parabolic density profile, the electric field was expressed as a linear combination of the normal modes for a quadratic profile. They found that the electric field of the beam displays aperiodic axial variation.

Instead of using wave optics to treat the problem, Steinhauer and Ahlstrom[10, 11] made a detailed study on the problem by means of geometrical optics. They considered the propagation of rays along a cylindrical plasma column with azimuthal symmetry. They found that rays would oscillate within a plasma column which has a parabolic density profile with an on axis minimum. The beam trapping phenomena were experimentally observed by several research groups[12, 13, 14].

Dudder and Henderson[15] developed a two dimensional (radial and axial) simulation model, RAMSES, for studying the propagation of a laser pulse through a background gas or plasma. Ray optics was used in the model. The ray trajectories were solved from the ray equation. Hubbard and Montes[16] also developed a program to trace the beam in a continuously varying refracting medium. They used the method of Taylor expansion about the initial position to approximate the new ray position. Acccording to the analysis, by knowing the refractive index of the medium, its gradient, the co-ordinates and tangent vector of an initial point on the ray, the co-ordinates of the ray path can be found. However, diffraction and phase information of the radiation were not included in the program and the initial positions and directions of the rays were chosen in a definite pattern.

Rinker and Bohannon[17] recently presented a description of the finite size focal spot of a laser beam. Rays were chosen to be normally distributed at the focal plane so that a Gaussian beam intensity profile was ensured at the spot. In the analysis, the rays are traced within a cylindrical symmetric medium by means of a two step procedure. The first step gives the ray positions at the boundaries of a plasma zone and the second step computes the ray path within the zone. Their overall treatment does not include the diffraction effects of the beam.

Tappert[11] developed a method to trace rays for a focused laser beam with the inclusion of diffraction effects. Formulae were derived for the spread of ray angles that yield the correct diffraction patterns for coherent and partially coherent beams. With this method, the size of the beam was found to reach the diffraction limit in the case of an ideal lens.

1.3 Purpose of present work

The purpose of the present work is to investigate the propagation of a focused laser beam in a plasma confined within a solenoidal magnetic field by using geometrical optics. The use of ray optics to describe beam propagation gives an alternate approach for obtaining the propagation behaviour of the beam. The objective can be divided into three parts:

1. *Description of a focused laser beam in vacuum using ray optics*

   By using rays to describe the focusing action of an ideal lens, one has to solve the problem of infinite intensity at the focus as all rays will merge into that point. Tappert suggested including the diffraction effects of a focused beam in the ray tracing technique so that a finite size focal spot can be attained. Based upon this idea, rays emerging from a converging lens are assigned a direction which deviate from the direction pointing at the focus. This deviation is chosen to follow a Gaussian distribution. With such chosen directions, rays will spread out at the focus in a Gaussian manner, avoiding the problem of infinite intensity.

2. *Beam propagation within a magnetically confined plasma column*

   The propagation of the beam within the plasma are studied in terms of ray optics. Past work has been done for the case of a parabolic density profile. The present objective is to trace rays within the plasma which has an arbitrary profile. This tracing is intended for implementing the magnetohydrodynamic(MHD) code for a plasma column developed by McMullin, Milroy and Capjack [19]. In the code, the spatial beam intensity profile and hence the laser power is assumed to be a constant of time within the column. This assumption does not include the effect of a changing refractive index which alters the beam profile. The tracing routine will be used to modify the laser power computation routine in the MHD code by including refractions. In this way, a more realistic simulation of the plasma can be obtained.

3. *Calculation of absorbed energy and ponderomotive forces*

Upon knowing how the beam propagates inside the column, energy deposition and absorption can be described more accurately. An energy absorption package is designed based upon the computed ray trajectories, since energy interchanges are taken place along the ray paths. Moreover, the MHD code is also designed for studying the magnetohydrodynamics for a gas target, the ponderomotive force becomes an important component affecting the dynamics of the plasma as high intensity laser beams are used in this case.

Discussions and derivations of the above objectives are presented in the following chapters. In chapter two, an explanation of Tappert's diffractive ray tracing technique is given. Here, rays can be chosen accordingly to simulate a focused laser beam propagating in vacuum. Cases for coherent and incoherent Gaussian beam are presented. The propagation of these rays inside the plasma is discussed in chapter three. Solutions to the ray trajectories in regions with different densities are derived. Chapter four gives a discussion on the energy absorption and ponderomotive forces within the plasma. Energy absorption is based on inverse Bremstrahlung process. Derivations of ponderomotive forces are based on the work of Chen[20]. The conversion of the analytical results obtained in previous chapters into numerical computations is presented in chapter five. Descriptions of program routines for generation of ray locations and directions, ray tracing and energy absorption are given and discussed. Results computed for a tested density profile within the plasma column are discussed in chapter six.

CGS units are adopted throughout the work, except power is given in watts.

Chapter 2

Diffractive ray tracing

The focusing action of an ideal converging lens can be determined by tracing rays according to the theory of geometrical optics. Ideally, rays parallel to the optical axis of a converging lens will all converge to the focus resulting with an infinite intensity at that point. However, according to the theory of diffraction[1], a light beam parallel to the optical axis of the lens is not focused to just a point but to a sizable area. The beam intensity at the geometrical focus is found to be

$$I = \left(\frac{\pi a^2 A}{\lambda f_L^2}\right)^2$$

where A is the amplitude of a spherical wave front, $\frac{Ae^{ikr}}{r}$ at unit distance from the source; f_L and a are focal length and aperture of lens respectively; λ is the wavelength of the incident beam. As can be seen from the expression, the focal intensity goes to infinity as the wavelength λ goes to zero which is the basic assumption used in geometrical optics.

A more accurate description of the lens focusing action can be provided by the ray tracing techniques if the diffraction effects are incorporated into the picture. This chapter discusses how rays can be chosen to take the diffraction effect into account by introducing the idea of a phase space distribution function. Based on this function, formulae for ray directions and for beam size are derived for a beam with a Gaussian amplitude and a coherent phase in the first case and an incoherent phase in the second case. Section 2.1 explains how the radiation energy of a beam can be distributed into particular locations and directions according to the above distribution function. This latter function for a Gaussian beam will be derived in section 2.2. Section 2.3 discusses how the distributed radiation energy can be assigned to rays with particular directions and locations. It also gives a description of the ray distribution at the lens and focal planes. Section 2.4 discusses the distribution function for a Gaussian beam with incoherent phases due to instrument limitations.

2.1 Tappert's phase-space distribution function

In this section, Tappert's phase space distribution function is defined. The significance of this function in the ray tracing technique is discussed. The energy density and energy flux of a laser beam are found to be the zero and first moments respectively of this function with respect to direction.

In conventional ray tracing of light propagation through a converging lens, each point at the lens plane can be associated with a ray directed towards the focus so that all rays will meet at one spot. In order to obtain the diffraction limited focus size, the directions of rays were chosen in such a way that not all the rays would intersect at the focus. Tappert's phase space distribution function provides a method of solving this problem.

Before defining the distribution function, a review on the derivation of energy flux and energy density from Helmoltz equation for the vacuum case is given in here for future reference.

The wave equation for a laser beam propagating in vacuum is given by the Helmoltz equation

$$\vec{\nabla}^2\vec{E} + k^2\vec{E} = 0$$

$$\vec{E} = \vec{\varepsilon}(\vec{r})\exp(i\vec{k}\cdot\vec{r} - i\omega t) \tag{2.1.1}$$

where $\vec{k}$ is the propagation vector in vacuum,and $|k|=\omega/c$; ω is the angular frequency of the wave; c is the speed of light; $\vec{r}$ is the position vector of a wavefront at a distance r from the source. Through assuming that the wave propagates primarily along the $\vec{z}$ direction, $\vec{E}$ is taken to have the form

$$\vec{E} = \vec{\varepsilon}(x,y,z)\exp(ikz - i\omega t) \tag{2.1.2}$$

By using the parabolic approximation (which states that the axial field amplitude variation over a wavelength is much smaller than the variation over the axial scale length,that is , ($\frac{\partial^2\varepsilon}{\partial z^2} << k\frac{\partial\varepsilon}{\partial z}$),eq. (2.1.1) is then reduced to

$$i\frac{\partial\varepsilon}{\partial z} + \frac{1}{2k}\frac{\partial^2\varepsilon}{\partial\vec{x}_\perp^2} = 0 \tag{2.1.3}$$

where

By taking the Fourier Transform of eq. (2.1.3) with respect to $\vec{k} = k_x\vec{i} + k_y\vec{j}$ and using the boundary conditions : (1)at infinity, the integrals of $\frac{\partial\varepsilon}{\partial x_\perp}$ and ε vanish over a surface perpendicular to the direction of propagation (see footnote) , the equation becomes

$$\frac{d\varepsilon_F}{dz} = -a^2 k_\perp^2 \varepsilon_F \tag{2.1.3a}$$

On solving eq. (2.1.3a) with the boundary condition that the electric field at the lens plane is $\varepsilon(x,y,z_0)$, the solution of the transformed equation is

$$\varepsilon_F(k_x,k_y,z) = \varepsilon_F(k_x,k_y,z_0)\exp[-a^2k_\perp^2(z-z_0)] \tag{2.1.4}$$

where

$$\varepsilon_F(k_x,k_y,z) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\exp(i\vec{k}_\perp\cdot\vec{x}_\perp)\varepsilon(\vec{x}_\perp,z)d^2\vec{x}_\perp$$

$$a^2 = \frac{1}{2ik} \tag{2.1.5}$$

By taking the inverse Fourier transform of eq. (2.1.4) and applying the convolution theorem, the amplitude of the electric field at $z>z_0$ is given by

$$\varepsilon(x,y,z) = \frac{k}{2\pi i(z-z_0)}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\varepsilon(\vec{x}_\perp',z_0)\exp[\frac{ik(\vec{x}_\perp-\vec{x}_\perp')^2}{2(z-z_0)}]d^2\vec{x}_\perp' \tag{2.1.6}$$

According to the electromagnetic theory, the time average energy density U and flux $\vec{S}$ of the radiation field which is characterized by eq. (2.1.2) are

$$U = \frac{1}{8\pi}|\varepsilon|^2 \tag{2.1.7}$$

$$\vec{S} = \frac{c}{8\pi}[|\varepsilon|^2\vec{z} - \frac{i}{2k}(\varepsilon^*\vec{\nabla}_\perp\varepsilon - \varepsilon\vec{\nabla}_\perp\varepsilon^*)] \tag{2.1.8}$$

where $\vec{\nabla}_\perp$ is $\frac{\partial}{\partial x}\vec{i} + \frac{\partial}{\partial y}\vec{j}$

These two quantities can be evaluated in terms of the integral derived in eq. (2.1.6). The

----

The boundary condition can be imposed because of the finiteness of field energy. At any time t, the wavefront from a source reaches a distance ct. If the boundary is taken to be further than ct, the field will be zero.

derivation of $\vec{S}$ is given in Appendix(1).

Instead of obtaining the energy density and flux by directly evaluating eq. (2.1.7) and eq. (2.1.8) with the electric field given by eq. (2.1.6), Tappert introduced the idea of using a phase space distribution function to evaluate the quantities. This distribution function was originated from Wigner[3] who introduced it in finding the probability of a particle having its location and momentum defined simultaneously.

Adopting this idea from Wigner[2,3], Tappert[4] was able to describe the propagation of beam energy in terms of a set of rays continuously distributed over space and directions by using a phase space distribution function in the form

$$f(\vec{x}_\perp,\vec{u}_\perp,z) = \frac{k^2}{(2\pi)^2}\int\int_{-\infty-\infty}^{\infty\infty} d^2\vec{x}_\perp' e^{ik\vec{u}_\perp\cdot\vec{x}_\perp'}\varepsilon(\vec{x}_\perp-\tfrac{1}{2}\vec{x}_\perp',z)\varepsilon^*(\vec{x}_\perp+\tfrac{1}{2}\vec{x}_\perp',z) \tag{2.1.9}$$

$\varepsilon(\vec{x}_\perp,z)$ is defined in eq. (2.1.2); $\vec{u}_\perp$ is a two dimensional unit vector whose x and y components denote the direction cosines of the ray direction (see fig. 2.1)

$$u_{\perp x} = \sin\theta\cos\phi$$

$$u_{\perp y} = \sin\theta\sin\phi$$

For rays propagating close to the axis of propagation, the magnitude of $\vec{u}_\perp$ is

$$|\vec{u}_\perp| = \sqrt{u_{\perp x}^2 + u_{\perp y}^2}$$

$$= |\sin\theta| = \theta$$

For small angle θ , the magnitude of $\vec{u}_\perp$ approximately equals to the angle which is the angle the ray subtends at the z-axis.

This function gives a description of the distribution of beam energy over space and direction. The amount of beam energy which is located at the point (x,y) and propagates in the direction $\vec{u}_\perp=\vec{u}_\perp+\vec{z}$ is determined from the function. The trajectory which is traced out by this pack of energy is a ray with defined origin $\vec{x}$ and direction $\vec{u}_\perp+\vec{z}$. Tappert's function is a real function and may take negative values. It has the following properties:

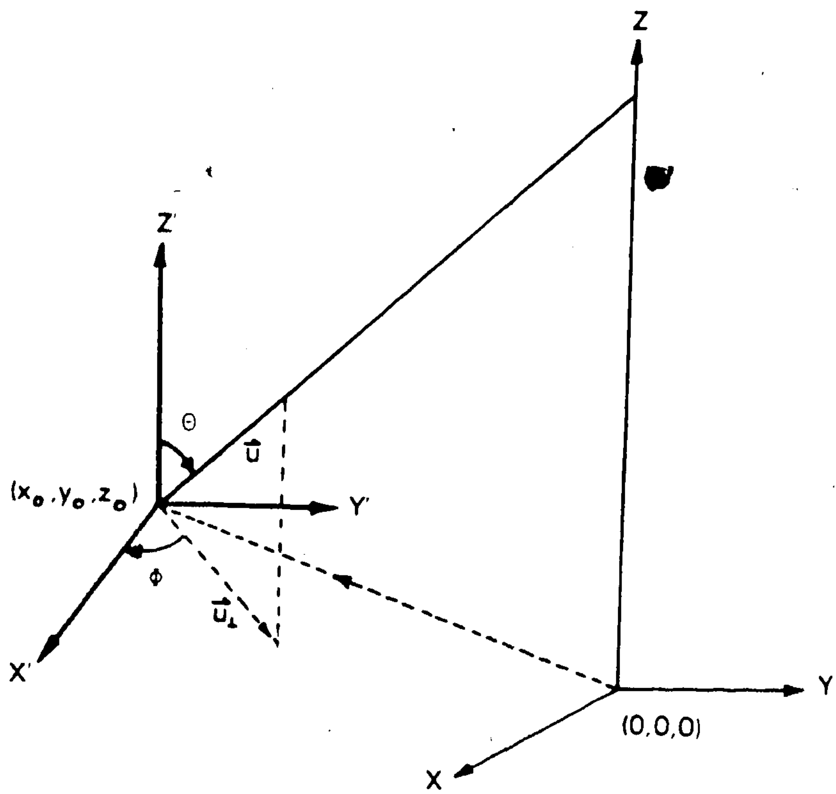


Figure 2.1 Co-ordinate system adopted for rays.

1. When it is integrated over all transverse directions $\vec{u}_\perp$ it gives the energy density over a volume element at location $(\vec{x}_\perp, z)$, that is,

$$U(\vec{x}_\perp, z) = \frac{1}{8\pi} \iint_{-1}^{1} f(\vec{x}_\perp, \vec{u}_\perp, z) d^2\vec{u}_\perp \qquad (2.1.10)$$

which after substituting f from eq. (2.1.9), gives

$$U(\vec{x}_\perp, z) = \frac{1}{8\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{k^2}{(2\pi)^2}d^2\vec{x}_\perp'\int_{-1}^{1}\int_{-1}^{1}e^{ik\vec{u}_\perp\cdot\vec{x}_\perp'}\varepsilon(\vec{x}_\perp-\tfrac{1}{2}\vec{x}_\perp', z)\varepsilon^*(\vec{x}_\perp+\tfrac{1}{2}\vec{x}_\perp', z)\, d^2\vec{u}_\perp$$

$$=\frac{1}{8\pi}\frac{k^2}{(\pi)^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{\sin kx'}{kx'}\frac{\sin ky'}{ky'}\varepsilon(\vec{x}_\perp-\tfrac{1}{2}\vec{x}_\perp', z)\varepsilon^*(\vec{x}_\perp+\tfrac{1}{2}\vec{x}_\perp', z)dx'dy'$$

Since the amplitude of the electric field is assumed to vary slowly over a wavelength, the field can be considered constant over the region of integration $(\frac{-\pi}{k}, \frac{\pi}{k})$. The value of the electric field at $\vec{x}_\perp'=0$ is taken to be the field value throughout the region $(\frac{-\pi}{k}, \frac{\pi}{k})$. Thus, the integral can be approximated as

$$U(\vec{x}_\perp, z) = \frac{1}{8\pi}\frac{k^2}{(\pi)^2}\left[\iint_{-\infty}^{\infty}\frac{\sin kx'}{kx'}\frac{\sin ky'}{ky'}\right]dx'dy'\varepsilon(\vec{x}_\perp, z)\varepsilon^*(\vec{x}_\perp, z)$$

$$= \frac{\varepsilon(\vec{x}_\perp, z)\varepsilon^*(\vec{x}_\perp, z)}{8\pi}\frac{k^2}{\pi^2}\iint_0^{\infty}\frac{\sin ky'}{ky'}\frac{4\sin kx'}{kx'}dx'dy'$$

$$= \frac{|\varepsilon(\vec{x}_\perp, z)|^2}{8\pi}\frac{k^2}{\pi^2}\frac{4\pi^2}{4k^2} = \frac{|\varepsilon(\vec{x}_\perp, z)|^2}{8\pi}$$

which is the energy density.

2. When $\vec{u}_\perp f$ is integrated with respect to $u_{\perp x}, u_{\perp y}$ the transverse flux components of the light beam along x , y directions is obtained

$$\vec{S}_\perp(\vec{x}_\perp, z) = \frac{c}{8\pi}\int_{-1}^{1}\int_{-1}^{1} f(\vec{x}_\perp, \vec{u}_\perp, z)\vec{u}_\perp d^2\vec{u}_\perp \qquad (2.1.11)$$

The aforemented statement is verified by substituting f into eq. (2.1.11) giving

$$\vec{S}_\perp(\vec{x}_\perp,z) = \frac{c}{8\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}dx'dy'\varepsilon(\vec{x}_\perp-\frac{1}{2}\vec{x}_\perp',z)\varepsilon^*(\vec{x}_\perp+\frac{1}{2}\vec{x}_\perp',z)$$

$$\times(\frac{k}{2\pi})^2\int_{-1}^{1}\int_{-1}^{1}(u_{\perp x}\vec{i}+u_{\perp y}\vec{j})e^{ik(u_{\perp x}x'+u_{\perp y}y')}du_{\perp x}du_{\perp y} \qquad (2.1.12)$$

Consider the x component of $S_\perp$.

$$S_x(\vec{x}_\perp,z) = \frac{c}{8\pi}\iint_{-\infty}^{\infty}dx'dy'\varepsilon(\vec{x}_\perp-\frac{1}{2}\vec{x}_\perp',z)\varepsilon^*(\vec{x}_\perp+\frac{1}{2}\vec{x}_\perp',z)$$

$$\times\frac{k^2}{(2\pi)^2}\iint_{-1}^{1}u_{\perp x}e^{ik(u_{\perp x}x'+u_{\perp y}y')}du_{\perp x}du_{\perp y}$$

$$= \frac{c}{8\pi}\frac{k^2}{(2\pi)^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}dx'dy'\varepsilon(\vec{x}_\perp-\frac{1}{2}\vec{x}_\perp',z)\varepsilon^*(\vec{x}_\perp+\frac{1}{2}\vec{x}_\perp',z)$$

$$\times\int_{-1}^{1}u_{\perp x}e^{iku_{\perp x}x'}du_{\perp x}\int_{-1}^{1}e^{iku_{\perp y}y'}du_{\perp y}$$

$$= \frac{c}{8\pi}(\frac{k}{2\pi})^2\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}dx'dy'\varepsilon(\vec{x}_\perp-\frac{1}{2}\vec{x}_\perp',z)\varepsilon^*(\vec{x}_\perp+\frac{1}{2}\vec{x}_\perp',z)$$

$$\times\frac{d}{dx'}\int_{-1}^{1}\frac{e^{iku_{\perp x}x'}}{ik}du_{\perp x}\frac{2\sin ky'}{ky'}$$

$$= \frac{c}{8\pi}(\frac{k}{2\pi})^2\frac{4}{ik}\int_{-\infty}^{\infty}\frac{d}{dx'}\frac{\sin kx'}{kx'}dx'$$

$$\times\int_{-\infty}^{\infty}\frac{\sin ky'}{ky'}\varepsilon(\vec{x}_\perp-\frac{1}{2}\vec{x}_\perp',z)\varepsilon^*(\vec{x}_\perp+\frac{1}{2}\vec{x}_\perp',z)dy'$$

By the same argument as in (i),

$$S_x(\vec{x}_\perp',z) = \frac{c}{8\pi}(\frac{k}{2\pi})^2\frac{4}{ik}\frac{\pi}{k}\int_{-\infty}^{\infty}\varepsilon(x-\frac{1}{2}x',y-\frac{1}{2}y',z)_{y'=0}$$

$$\times\varepsilon^*(x+\frac{1}{2}x',y+\frac{1}{2}y',z)_{y'=0}[\frac{d}{dx'}\frac{\sin kx'}{kx'}]dx'$$

$$= \frac{c}{8\pi}\left(\frac{k}{2\pi}\right)^2 \frac{4\pi}{ik^2}\left[-\int_{-\infty}^{\infty}\frac{\sin kx'}{kx'}\frac{\partial}{\partial x'}\varepsilon\varepsilon^* dx'\right]$$

Since the field is assumed to vary slowly in space, the derivative of the product of the field amplitudes, $\frac{\partial}{\partial x'}(\varepsilon\varepsilon^*)$ can be considered constant and taken to be the value at x'=0. Thus, after integrating,

$$S_x(\vec{x}_\perp, z) = \frac{c}{8\pi}\frac{1}{ik}\frac{\partial}{\partial x'}(\varepsilon\varepsilon^*)\Big|_{x'=0}$$

Similarly, for the y component of $\vec{S}$,

$$S_y(\vec{x}_\perp, z) = \frac{c}{8\pi}\frac{1}{ik}\frac{\partial}{\partial y'}(\varepsilon\varepsilon^*)\Big|_{y'=0}$$

By adding the x and y components, the transverse flux component is given by

$$\vec{S}_\perp = S_x\vec{i} + S_y\vec{j}$$

$$= \frac{c}{8\pi}\frac{1}{ik}\vec{\nabla}'_\perp(\varepsilon\varepsilon^*)\Big|_{x'=0,y'=0}$$

By using the transform of the co-ordinates,

$$\vec{\nabla}'_\perp\varepsilon(\vec{x}_\perp - \tfrac{1}{2}\vec{x}'_\perp, z) = -\tfrac{1}{2}\vec{\nabla}_\perp\varepsilon(\vec{x}_\perp - \tfrac{1}{2}\vec{x}'_\perp, z) \qquad (2.1.12a)$$

$$\vec{\nabla}'_\perp\varepsilon^*(\vec{x}_\perp + \tfrac{1}{2}\vec{x}'_\perp, z) = \tfrac{1}{2}\vec{\nabla}_\perp\varepsilon^*(\vec{x}_\perp + \tfrac{1}{2}\vec{x}'_\perp, z) \qquad (2.1.12b)$$

$$\vec{\nabla}^2_\perp = 4\vec{\nabla}'^2_\perp \qquad (2.1.13)$$

the transverse flux vector becomes

$$\vec{S}_\perp = \frac{c}{8\pi}\frac{i}{2k}\left[\varepsilon\vec{\nabla}_\perp\varepsilon^* - \varepsilon^*\vec{\nabla}_\perp\varepsilon\right]_{x'=0,y'=0} \qquad (2.1.14)$$

The variation of the distribution function along a trajectory can be found by differentiating eq. (2.1.9) with respect to z, that is,

$$\frac{\partial f}{\partial z} = \frac{k^2}{(2\pi)^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} d^2\vec{x}_\perp' e^{ik\vec{u}_\perp\cdot\vec{x}_\perp'}\left(\varepsilon\frac{\partial\varepsilon^*}{\partial z} + \varepsilon^*\frac{\partial\varepsilon}{\partial z}\right) \qquad (2.1.15)$$

Multiply eq. (2.1.3) by ε^* and its complex conjugate by ε ,

$$i\varepsilon^*\frac{\partial\varepsilon}{\partial z} + \frac{\varepsilon^*}{2k}\vec{\nabla}_\perp^2\varepsilon = 0 \qquad (2.1.16)$$

$$-i\varepsilon\frac{\partial\varepsilon^*}{\partial z} + \frac{\varepsilon}{2k}\vec{\nabla}_\perp^2\varepsilon^* = 0 \qquad (2.1.17)$$

eq. (2.1.15) becomes

$$\frac{\partial f}{\partial z} = \frac{1}{2ik}\frac{k^2}{(2\pi)^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} d^2\vec{x}_\perp' e^{ik\vec{u}_\perp\cdot\vec{x}_\perp'}\left(\varepsilon\vec{\nabla}_\perp^2\varepsilon^* - \varepsilon^*\vec{\nabla}_\perp^2\varepsilon\right) \qquad (2.1.18)$$

And from eq. (2.1.12) and eq. (2.1.13), $\frac{\partial f}{\partial z}$ becomes

$$\frac{\partial f}{\partial z} = \frac{2}{ik}\left(\frac{k}{2\pi}\right)^2\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} d^2\vec{x}_\perp' e^{ik\vec{u}_\perp\cdot\vec{x}_\perp'}\left(\varepsilon\vec{\nabla}_\perp'^2\varepsilon^* - \varepsilon^*\vec{\nabla}_\perp'^2\varepsilon\right) \qquad (2.1.19)$$

Integrating each term by parts and using the boundary condition $\varepsilon(x,y,z)\big|_{x\to-\infty,\ y\to-\infty} = 0$, one gets

$$\frac{\partial f}{\partial z} = \frac{k^2}{(2\pi)^2}\left(\frac{2}{ik}\right)\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\vec{\nabla}_\perp'\left(\varepsilon^* e^{ik\vec{u}_\perp\cdot\vec{x}_\perp'}\right)\vec{\nabla}_\perp'\varepsilon\, d^2\vec{x}_\perp' - \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\vec{\nabla}_\perp'\left(\varepsilon e^{ik\vec{u}_\perp\cdot\vec{x}_\perp'}\right)\vec{\nabla}_\perp'\varepsilon^* d^2\vec{x}_\perp'\right] \qquad (2.1.20)$$

Expanding the integrand,

$$\frac{\partial f}{\partial z} = 2\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\frac{k}{2\pi}\right)^2\left[\varepsilon^*\vec{u}_\perp\cdot\vec{\nabla}_\perp'\varepsilon - \vec{u}_\perp\cdot\vec{\nabla}_\perp'\varepsilon^*\right]e^{ik\vec{u}_\perp\cdot\vec{x}_\perp'}d^2\vec{x}_\perp' \qquad (2.1.21)$$

From eq. (2.1.12a) and eq. (2.1.12b), replace $\vec{\nabla}_\perp'$ by $\vec{\nabla}_\perp$.

$$\frac{\partial f}{\partial z} = \frac{-k^2}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} d^2\vec{x}_\perp'\left[\varepsilon^*(\vec{u}_\perp\cdot\vec{\nabla}_\perp\varepsilon) + \varepsilon(\vec{u}_\perp\cdot\vec{\nabla}_\perp\varepsilon^*)\right]e^{ik\vec{u}_\perp\cdot\vec{x}_\perp'}$$

$$= \frac{-k^2}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(\vec{u}_\perp\cdot\vec{\nabla}_\perp\varepsilon\varepsilon^*)e^{ik\vec{u}_\perp\cdot\vec{x}_\perp'}d^2\vec{x}_\perp'$$

or

$$\frac{\partial f}{\partial z} = -\vec{u}_\perp\cdot\vec{\nabla}_\perp f \qquad (2.1.22)$$

Using the method of characteristics in partial differential equations to solve eq (2.1.22), the corresponding characteristic equations for eq (2.1.22) are

$$\frac{dx}{dt} = u_{\perp x}$$

$$\frac{dy}{dt} = u_{\perp y}$$

$$\frac{dz}{dt} = 1 \tag{2.1.23}$$

where t is a parameter. On integrating, the solution to eq. (2.1.23) is,

$$x = u_{\perp x}t + c_1$$

$$y = u_{\perp y}t + c_2$$

$$z = t + c_3 \tag{2.1.24}$$

This set of equations denotes a parametric curve on which eq. (2.1.22) is satisfied. In other words, the value of f remains constant along this set of curves.

By using the initial conditions for x, y, z, namely, at t=0, $x=x_0$, $y=y_0$, $z=z_0$, the family of curves can be written as

$$x = x_0 + u_{\perp x}t$$

$$y = y_0 + u_{\perp y}t$$

$$z = z_0 + t \tag{2.1.25}$$

Through replacing t with $z-z_0$ from the last equation of (2.1.25), the above equations become

$$x = x_0 + u_{\perp x}(z-z_0)$$

$$y = y_0 + u_{\perp y}(z-z_0) \tag{2.1.26}$$

Or, in vector notation,

$$\vec{x} = \vec{x}_0 + \vec{u}_\perp(z-z_0) \tag{2.1.27}$$

Since according to eq. (2.1.22), f remains unchanged along the path, one can write

$$f(\vec{x}_{\perp 0}+\vec{u}_{\perp}(z-z_0),\vec{u}_{\perp},z) = f(\vec{x}_{\perp 0},\vec{u}_{\perp}, z_0) \tag{2.1.28}$$

By substituting for the arguments of f from eq. (2.1.26), eq. (2.1.28) becomes

$$f(\vec{x}_{\perp}, \vec{u}_{\perp}, z) = f(\vec{x}_{\perp}-\vec{u}_{\perp}(z-z_0), \vec{u}_{\perp}, z_0) \tag{2.1.29}$$

The path given by eq. (2.1.26) may be regarded as a trajectory for energy packets with the same initial positions $\vec{x}_{\perp 0}$ and the same initial direction $\vec{u}_{\perp}$.

2.2 Tappert's distribution function of a coherent Gaussian beam

In this section, the phase space distribution function for a coherent Gaussian beam is derived. From this distribution function, the average directions of the rays, beam energy density and spot size are deduced.

Under the far field approximation, the wave amplitude of a collimated Gaussian laser beam before passing through a converging lens at a distance z_0 from the lens[5], is given by,

$$\epsilon(x,y,z_0) = \sqrt{\frac{2}{\pi}}\frac{1}{w(z_0)}e^{i(kz_0-\phi)}e^{-r^2/w^2(z_0)}e^{-ikr^2/2R(z_0)} \tag{2.2.1}$$

where

$$\phi = \tan^{-1}\left(\frac{\lambda z_0}{\pi w_0^2}\right)$$

$$R(z_0) = z_0\left[1+\left(\frac{\pi w_0^2}{\lambda z_0}\right)^2\right]$$

$$w^2(z_0) = w_0^2\left[1+\left(\frac{\lambda z_0}{\pi w^2}\right)^2\right]$$

$$r^2 = x^2 + y^2$$

$r \ll R$ and w_0 is the beam waist of the laser.

On passing through the lens which is assumed to be thin, the field amplitude just after the lens($z=z_1$), becomes(see Appendix 2)

$$\epsilon(x,y,z_1) = E_0 e^{-r^2/a_0^2}\, e^{-ikr^2/2f_L} \tag{2.2.2}$$

where E_0 is

$$\sqrt{\frac{2}{\pi}}\frac{1}{a_0}e^{-i(kz_0-\phi)}$$

a_0 is $w(z_0)$; f_L is the focal length of the lens.

By substituting this expression into eq. (2.1.9), the distribution function at a particular point (x_0,y_0,z_1) becomes

$$f(x_0,y_0,u_x,u_y,z_1) = \frac{|E_0|^2 k^2 a_0^2}{2\pi} e^{-2r_0^2/a_0^2} e^{-\frac{k^2a_0^2}{2}(u_x+\frac{x_0}{f_L})^2} e^{-\frac{k^2a_0^2}{2}(u_y+\frac{y_0}{f_L})^2}$$

(2.2.3)

At any particular location on x−y plane(constant r), the function has a maximum value at

$$u_x = -\frac{x_0}{f_L}$$

$$u_y = -\frac{y_0}{f_L}$$

(2.2.4)

This shows that most of the radiation energy at location(x_0, y_0) is associated with the ray pointing towards the focus (see fig. 2.2), while the rest of the radiation energy will be spread around this ray according to the distribution function given in eq. (2.2.3). By taking the first moment of $f(x_0,y_0,u_x,u_y,z_1)$, the average direction of the rays at the point (x_0,y_0,z_1) is

$$< u_x > = \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u_x f(x_0,y_0,u_x,u_y,z_1)du_x du_y}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_0,y_0,u_x,u_y,z_1)du_x du_y} = \frac{-x_0}{f_L}$$

$$< u_y > = \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u_y f(x_0,y_0,u_x,u_y,z_1)du_x du_y}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_0,y_0,u_x,u_y,z_1)du_x du_y} = \frac{-y_0}{f_L}$$

(2.2.5)

which is the peak value of $f(x_0,y_0,u_x,u_y,z_1)$. Thus,the average direction of the rays is the direction along which most rays will follow. Moreover, this direction is the same as that for a single ray emerging from the point (x_0,y_0) passing through the focus. Thus, the beam focusing action can be described in terms of a collection of rays.

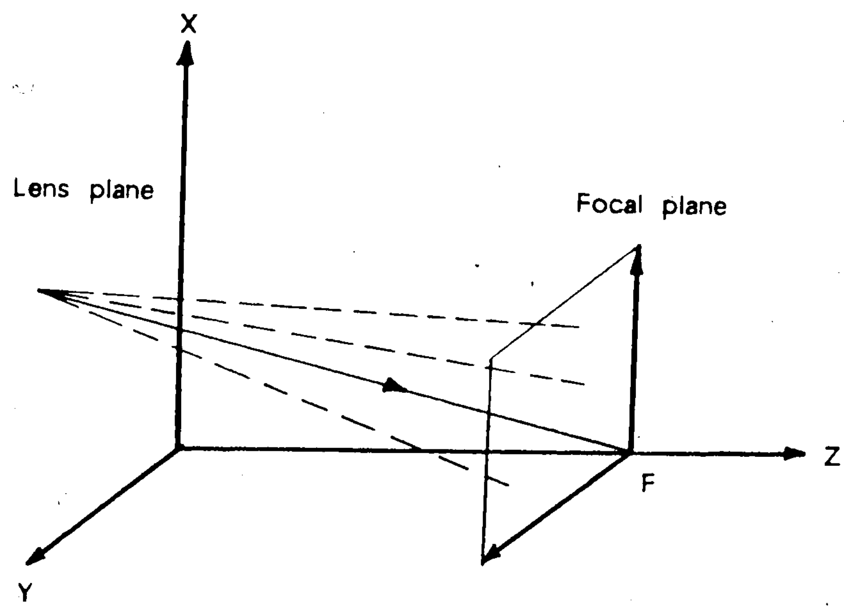


Figure 2.2 Spread of rays at focal plane.

By using eqs.(2.1.10), (2.1.11), the energy density and transverse flux components of the beam at a point (x_0, y_0) are given as,

$$U(x_0, y_0, z_1) = \frac{1}{8\pi} \frac{|E_0^2| k^2 a_0^2}{2\pi} e^{-2r_0^2/a_0^2} \times \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{\frac{-k^2 a_0^2}{2}\left[(u_x + \frac{x_0}{f_L})^2 + (u_y + \frac{y_0}{f_L})^2\right]} du_x du_y$$

$$= \frac{|E_0|^2 e^{-2r_0^2/a_0^2}}{8\pi} \tag{2.2.6}$$

$$\vec{S}_\perp(x_0, y_0, z_1) = \frac{c}{8\pi} \frac{|E_0|^2 k^2 a_0^2}{2\pi} e^{-2r_0^2/a_0^2} \times \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{\frac{-k^2 a_0^2}{2}\left[(u_x + \frac{x_0}{f_L})^2 + (u_y + \frac{y_0}{f_L})^2\right]} \times (u_x \vec{i} + u_y \vec{j}) du_x du_y$$

$$= \frac{c|E_0|^2}{8\pi}\left(\frac{-x_0}{f_L}\vec{i} - \frac{y_0}{f_L}\vec{j}\right) e^{-2r_0^2/a_0^2} \tag{2.2.7}$$

The distribution function from the lens plane onward can be derived by substituting eq. (2.1.26) into eq. (2.2.3)

$$f(x, y, u_x, u_y, z) = \frac{|E_0|^2 k^2 a_0^2}{2\pi} e^{\frac{-2[(x - u_x(z - z_0))^2]}{a_0^2}} \times e^{\frac{-2(y - u_y(z - z_0))^2}{a_0^2}} e^{\frac{-k^2 a_0^2}{2}\left(u_x + \frac{x - u_x(z - z_0)}{f_L}\right)^2} \times e^{\frac{-k^2 a_0^2}{2}\left(u_y + \frac{y - u_y(z - z_0)}{f_L}\right)^2} \tag{2.2.8}$$

Expanding the exponent and combining terms in x^2, y^2, u_x, u_y, eq. (2.2.8) becomes

$$f(x,y,u_x,u_y,z) = \frac{|E_0|^2k^2a_0^2}{2} e^{\frac{-2(x^2+y^2)}{a^2(z)}} e^{\frac{-k^2a^2(z)}{2}(u_x+\frac{x}{F(z)})^2} \times e^{\frac{-k^2a^2(z)}{2}(u_y+\frac{y}{F(z)})^2} \quad (2.2.9)$$

where

$$a^2(z) = a_0^2[(1-\frac{z-z_0}{f_L})^2+ \frac{4(z-z_0)^2}{k^2a_0^4}] \quad (2.2.10)$$

$$F(z) = f_L[\frac{(1-\frac{z-z_0}{f_L})^2+\frac{4(z-z_0)^2}{k^2a_0^4}}{1-\frac{z-z_0}{f_L}(1+\frac{4f_L^2}{k^2a_0^4})}] \quad (2.2.11)$$

By substituting for f from eq. (2.2.9) into eq. (2.2.5),the average value for $u_x u_y$ are evaluated to be

$$< u_x > = \frac{-x}{F(z)} \quad (2.2.12)$$

$$< u_y > = \frac{-y}{F(z)} \quad (2.2.13)$$

Moreover,by similar integration procedure as in eq. (2.2.6), and eq. (2.2.7), the energy density and transverse flux components are found to be

$$U(\vec{x}_\perp,z) = \frac{|E_0|^2}{8\pi} e^{-2r^2/a^2(z)} \quad (2.2.14)$$

$$\vec{S}_\perp(\vec{x}_\perp,z) = \frac{c}{8\pi}|E_0|^2e^{-2r^2/a^2(z)}[\frac{-x}{F(z)}\vec{i} - \frac{y}{F(z)}\vec{j}] \quad (2.2.15)$$

where $r^2=x^2+y^2$.

From eq. (2.2.10) the minimum spotsize is found to be

$$a_{min}^2 = \frac{4f_L^2/k^2a_0^2}{1 + \frac{4f_L^2}{k^2a_0^4}} \quad (2.2.16)$$

at an axial distance

$$z = z_0 + \frac{f_L}{1 + \frac{4f_L^2}{k^2 a_0^4}}. \tag{2.2.17}$$

From such a distribution function, the beam size at the focus reaches a limit which is governed by the wavelength of the beam. Moreover, the beam is focused within a focal region with radius a_{min} located at a distance slightly less than the focal length of the lens.

2.3 Ray description of a coherent Gaussian beam

In this section, Tappert's distribution function is represented as a sum of energy packages. Each of these packages is associated with a particular direction and location, determined from a sampling of Tappert's distribution function. These sampled values give the initial directions and locations of the rays.

As shown in the previous section, the energy flux of a laser beam can be evaluated from the first moment of the distribution function. This evaluation involves an integration over a continuous variation of transverse directions and positions. However, in using rays to trace the propagation of beam flux, a finite set of discrete directions and locations is chosen since only a finite number of rays can be used for describing the beam. These rays provide trajectories for the propagation of energy flux (macrophotons) (see footnote). The total energy flux over an area can be found by summing over all the macrophotons passing through that area.

Based on this idea, the distribution function is expressed in terms of a sum of macrophotons defined in particular locations and directions. The distribution function at the lens plane can be represented as

---

The term,macrophotons,are used in the work of Dudder and Henderson. The term represents a group of photons.

$$f(\vec{x}_\perp, \vec{u}_\perp, z_1) = \sum_{j=1}^{N} \omega_j \delta^2(\vec{x}_\perp - \vec{x}_{\perp j})\delta^2(\vec{u}_\perp - \vec{u}_{\perp j}) \tag{2.3.1}$$

where $\vec{x}_\perp$ is the position vector of a macrophoton; $\vec{u}_\perp$ is the direction vector of the macrophoton; ω_j is the amount of energy carried by the macrophoton; N is the number of macrophotons describing the laser beam; δ^2 is a two dimensional delta function. By substituting eq. (2.3.1) for f in eq. (2.1.10), the energy density is found to be

$$\begin{aligned} U(\vec{x}_\perp, z_1) &= \frac{1}{8\pi} \int f(\vec{x}_\perp, \vec{u}_\perp, z_1)\, d^2\vec{u}_\perp \\ &= \frac{1}{8\pi} \sum_{j=1}^{N} \omega_j \delta^2(\vec{x}_\perp - \vec{x}_{\perp j}) \int \delta^2(\vec{u}_\perp - \vec{u}_{\perp j})\, d^2\vec{u}_\perp \\ &= \frac{1}{8\pi} \sum_{j=1}^{N} \omega_j \delta^2(\vec{x}_\perp - \vec{x}_{\perp j}) \end{aligned} \tag{2.3.2}$$

The power flowing across an area ΔA along the direction of propagation is

$$P = \int_{\Delta A} \vec{S} \cdot \vec{z}\, d^2\vec{x}_\perp = \int_{\Delta A} S_z\, dxdy$$

where $\vec{S} = \vec{S}_\perp + \vec{S}_z$ and $\vec{z}$ is the unit vector along the direction of propagation. From Appendix 1, the z-component of $\vec{S}$ is

$$S_z = cU(\vec{x}_\perp, z_1)$$

Therefore, the power is given by

$$P = \int_{\Delta A} cU(\vec{x}_\perp, z_1)\, dx\, dy$$

From eq. (2.3.2),

$$\begin{aligned} P &= c \int_{\Delta A} \frac{1}{8\pi} \sum_{j=1}^{N} \omega_j \delta^2(\vec{x}_\perp - \vec{x}_{\perp j})\, dx\, dy \\ &= \frac{c}{8\pi} \sum_{j=1}^{N'} \omega_j \end{aligned} \tag{2.3.3}$$

where N' denotes those macrophotons lying within an area ΔA.

As discussed previously, rays emerge from each point on the lens plane as a bundle. Each of these rays will be assigned a particular direction. The choice of these directions is based on a Gaussian distribution of the directions of rays around the average direction of the ray bundle. As will be shown next, for such a choice, the beam intensity profile at the focal plane will also follow a Gaussian distribution.

Let U be a random variable representing the set of transverse directions of rays. Recalling from eq. (2.2.5) that the average transverse direction of a set of rays emerging from a point (x_0, y_0) at the lens plane is

$$< U_\perp > = \frac{-(x_0\vec{i} + y_0\vec{j})}{f_L} \qquad (2.3.4)$$

For each ray, the direction will be

$$\vec{u}_\perp = < U_\perp > + \Delta\vec{u}_\perp \qquad (2.3.5)$$

where $\Delta\vec{u}_\perp$ is the direction deviation from the average value. By combining eq. (2.3.4) and eq. (2.3.5), the ray direction is given by

$$\vec{u}_\perp = \frac{-(x_0\vec{i} + y_0\vec{j})}{f_L} + \Delta\vec{u}_\perp \qquad (2.3.6)$$

The location of such a ray at the focal plane will be

$$\begin{aligned} \vec{x}_\perp &= \vec{u}_\perp f_L + x_0\vec{i} + y_0\vec{j} \\ &= \Delta\vec{u}_\perp f_L \end{aligned} \qquad (2.3.7)$$

According to eq. (2.3.7), the ray location is only proportional to the deviation of the ray direction from the average value. For a ray with an average direction $<U_\perp>$, it crosses the axis of propagation at the focal plane. Rays with directions other than this will resume a location different from the origin. Since the number of rays decreases in a Gaussian manner with respect to the deviation of directions, the number of rays which resume locations away from the origin will consequently vary in a Gaussian manner.

2.4 Ray description of an incoherent Gaussian beam

In this section, the phase space distribution function and the formula for the beam width of an incoherent Gaussian beam is derived. The expression for the distribution function is found to be very similar to that for the coherent case except for a coherence factor. This factor accounts for the degree of coherence of the electric fields at different locations at the lens plane.

In practice, the beam can hardly be focussed to the diffraction limited spotsize due to instrumental limitations and imperfections. According to Tappert, these defects can be incorporated into the phase fluctuations of the electric field amplitudes at the lens plane. With such phase fluctuations, the field diffraction pattern at the focus changes. The energy distribution at the focus and accordingly, the spotsize can then be varied.

Phase fluctuations are accounted for by considering an ensemble (see footnote) of fields at the lens plane. Each member of the ensemble consists of electric fields over the plane transverse to the beam. A random phase is associated with the electric field at each location on the plane. As a result, the electric field amplitude used in eq. (2.1.9) is replaced by the average of the ensemble of field amplitudes. Upon substitution, the phase space distribution function then becomes

$$< f(\vec{x}_\perp, \vec{u}_\perp, z) > = \frac{k^2}{(2\pi)^2} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} d^2\vec{x}_\perp' e^{ik\vec{u}_\perp\cdot\vec{x}_\perp'} < \varepsilon(\vec{x}_\perp - \tfrac{1}{2}\vec{x}_\perp', z)\varepsilon^*(\vec{x}_\perp + \tfrac{1}{2}\vec{x}_\perp', z) > \qquad (2.4.1)$$

where the brackets, < >, denote ensemble averages of the bracketed quantity. The energy density and flux of the field are respectively

$$< U > = \frac{1}{8\pi}\int_{-1}^{1}\int_{-1}^{1} < f(\vec{x}_\perp, \vec{u}_\perp, z) > d^2\vec{u}_\perp = \frac{<|\varepsilon|^2>}{8\pi} \qquad (2.4.2)$$

$$< \vec{S}_\perp > = \frac{c}{8\pi}\int_{-1}^{1}\int_{-1}^{1} \vec{u}_\perp < f(\vec{x}_\perp, \vec{u}_\perp, z) > d^2\vec{u}_\perp$$
$$= \frac{c}{8\pi}\frac{i}{2k} < (\varepsilon\vec{\nabla}_\perp\varepsilon^* - \varepsilon^*\vec{\nabla}_\perp\varepsilon) > \qquad (2.4.3)$$

By applying this idea of random phase to a Gaussian beam, the electric field amplitude at the lens plane will be given by

$$\varepsilon(x, y, z_1) = E_0 e^{-r^2/a_0^2} e^{-ikr^2/2f_L} e^{i\Psi(x,y)}$$
$$= \varepsilon_1 e^{i\Psi(x,y)} \qquad (2.4.4)$$

where $\Psi(x, y)$ is the random phase of the electric field at the location (x,y); ε_1 is the

---

An ensemble is a collection of identical systems quantity $E_0 e^{\frac{-r^2}{2a_0^2}} e^{\frac{-ikr^2}{2f_L}}$. Although the phase values are regarded as random, they are assumed to be Gaussianly correlated in a small area with radius L within the beam cross sectional area. Within such a region, the phase value at any point (x,y) is given by

$$\psi(x,y) = \psi_0 e^{\frac{-(x-x_c)^2+(y-y_c)^2}{L^2}} \tag{2.4.5}$$

where (x_c,y_c) is the location at which the phase value is maximum. The phase values are assumed to be isotropic in the sense that the values within a localized area of radius L do not change regardless of the location of the area within the beam. Under such assumption, for any two points within the beam, the correlation function between the phase values is given by

$$\rho(x_2-x_1,y_2-y_1) = \frac{2}{\psi_0^2\pi L^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}dx_c dy_c \psi_0^2 \, e^{-\frac{(x_1-x_c)^2+(y_1-y_c)^2}{L^2}} e^{-\frac{(x_2-x_c)^2+(y_2-y_c)^2}{L^2}}$$

$$= e^{\frac{-1}{2L^2}[(x_2-x_1)^2 + (y_2-y_1)^2]}$$

$$= e^{-r'^2/2L^2} \tag{2.4.6}$$

where $r' = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

The correlation function for the phases at any two points is readily seen from eq. (2.4.6) to be dependent only upon the distance between them.

However, this correlation function can be formulated according to the concept of ensemble average as follows. Let the continuous range of possible values of Ψ be ψ_1 ψ_2, ψ_3 ,etc. The phase for the electric field in each member of the ensemble can take any of the values. Consider the phase values of each member at any two particular points (x_1,y_1) and (x_2,y_2). Let Φ_1 be the random variable to denote the set of phase values at (x_1,y_1), Φ_2 to denote those at (x_2,y_2). Since the phase values are restricted only to the set of ψ values mentioned, the values of Φ_1 and Φ_2 will lie within the range of the ψ values as well. A joint probability distribution function, $P_{\Phi_1\Phi_2}(\psi_1,\psi_2;x_2-x_1,y_2-y_1)$ can then be set up for the product of the random variables $\Phi_1\Phi_2$. ψ_1, ψ_2 are the values taken by Φ_1, Φ_2 respectively;and (x_2-x_1), (y_2-y_1) represent the spatial difference between the two points. Thus, the ensemble average of $\phi_1\phi_2$ or $<\phi_1\phi_2>$ can be calculated in terms of this distribution function as follows:

$$<\phi_1\phi_2> = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\psi_1\psi_2 P_{\phi_1\phi_2}(\psi_1,\psi_2;x_2-x_1,y_2-y_1)d\psi_1 d\psi_2 \quad (2.4.7)$$

where $\psi^2 = <\phi_1^2>$ or $<\phi_2^2>$ because ϕ_1 and ϕ_2 have the same kind of distribution. Since as it was mentioned earlier, the range of ϕ values is the same as that for Ψ, the average of the product $\Psi(x_1, y_1)\Psi(x_2, y_2)$ for a member of the ensemble is the same as the ensemble average. From eq. (2.4.6), the average of $\Psi(x_1, y_1)\Psi(x_2, y_2)$ over all points within the beam is the correlation function for the phase values at (x_1,y_1) and (x_2,y_2). Thus, the ensemble average of $\phi_1\phi_2$ is equivalent to the correlation function, that is,

$$\rho(x_2-x_1,y_2-y_1) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\psi_1\psi_2 P_{\phi_1\phi_2}(\psi_1,\psi_2;x_2-x_1,y_2-y_1)d\psi_1 d\psi_2 \quad (2.4.8)$$

The distribution functions for ϕ_1 and ϕ_2 are chosen to be Gaussian so that Tappert's distribution function can be simplified. The joint probability distribution function for $\phi_1\phi_2$ has the following form:

$$P_{\phi_1\phi_2}(\psi_1,\psi_2;x_2-x_1,y_2-y_1)$$

$$= \frac{e^{\frac{-1}{2[1-\rho^2(x_2-x_1,y_2-y_1)]}\left[\frac{\psi_1^2}{\psi_0^2} - 2\rho(x_2-x_1,y_2-y_1)\frac{\psi_1\psi_2}{\psi_0^2} + \frac{\psi_2^2}{\psi_0^2}\right]}}{2\pi\psi_0^2\sqrt{1-\rho^2(x_2-x_1,y_2-y_1)}} \quad (2.4.9)$$

The marginal probability, $P_{\phi_1}(\psi_1)$ will be

$$\sim p_{\phi_1}(\psi_1) = \int_{-\infty}^{\infty} p_{\phi_1\phi_2}(\psi_1,\psi_2;x_2-x_1,y_2-y_1)d\psi_2$$

$$= \frac{\int_{-\infty}^{\infty} e^{\frac{-[\frac{\psi_1^2}{\psi_0^2} - \frac{2\rho(x_2-x_1,y_2-y_1)\psi_1\psi_2}{\psi_0^2} + \frac{\psi_2^2}{\psi_0^2}]}{2[1-\rho^2(x_2-x_1,y_2-y_1)]}} d\psi_2}{2\pi\psi_0^2\sqrt{1-\rho^2(x_2-x_1,y_2-y_1)}}$$

$$= \frac{\int_{-\infty}^{\infty} e^{\frac{-1}{2\psi_0^2}[\psi_1^2 + \frac{(\rho(x_2-x_1,y_2-y_1)\psi_1 - \psi_2)^2}{1-\rho^2(x_2-x_1,y_2-y_1)}]} d\psi_2}{2\pi\psi_0^2\sqrt{1-\rho^2(x_2-x_1,y_2-y_1)}}$$

$$= \frac{e^{-\psi_1^2/2\psi_0^2}}{\psi_0\sqrt{2\pi}} \tag{2.4.10}$$

Hence the set of phase values represented by the random variable $\phi_1(x, y)$ follows a Gaussian distribution as expected. This Gaussian property of ϕ_1 gives the average value of $e^{i\phi_1}$ to be

$$< e^{i\phi_1} > = \frac{1}{\sqrt{2\pi}\psi_0}\int_{-\infty}^{\infty} e^{-\psi_1^2/2\psi_0^2} e^{i\psi_1} d\psi_1$$

$$= e^{-\psi_0^2/2} = e^{\frac{-1}{2}<\phi_1^2>} \tag{2.4.11}$$

From eq. (2.4.4) and eq. (2.1.9), for a member of the ensemble the phase space distribution function is

$$f(\vec{x}_\perp,\vec{u}_\perp,z) = \frac{k^2}{(2\pi)^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} d^2\vec{x}_\perp' e^{ik\vec{u}_\perp\cdot\vec{x}_\perp'}\epsilon_1(\vec{x}_\perp - \tfrac{1}{2}\vec{x}_\perp',z)\epsilon_1^*(\vec{x}_\perp + \tfrac{1}{2}\vec{x}_\perp',z)$$

$$\times e^{i[\psi(\vec{x}_\perp - \frac{1}{2}\vec{x}_\perp') - \psi(\vec{x}_\perp + \frac{1}{2}\vec{x}_\perp')]} \tag{2.4.12}$$

Moreover, the ensemble average of $f(\vec{x}_\perp,\vec{u}_\perp,z)$ may be derived as follows. Since the range of values of ψ is always the same for any point in x–y plane, the values of $\psi(\vec{x}_\perp - \frac{1}{2}\vec{x}_\perp')$

and $\psi(\vec{x}_\perp + \frac{1}{2}\vec{x}'_\perp)$ can be represented by ψ_1 and ϕ_2 respectively. In this case, the joint probability distribution function given previously becomes $P_{\phi_1\phi_2}(\psi_1, \psi_2; x', y')$. The ensemble average of $< e^{i(\phi_1-\phi_2)} >$ is given by

$$< e^{i(\phi_1-\phi_2)} > = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{i(\psi_1-\psi_2)} P_{\phi_1\phi_2}(\psi_1, \psi_2; x', y')\, d\psi_1 d\psi_2 \tag{2.4.13}$$

Substituting for $P_{\phi_1\phi_2}(\psi_1, \psi_2; x', y')$ from eq. (2.4.9),

$$< e^{i(\phi_1-\phi_2)} > = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{e^{i(\psi_1-\psi_2)}\, e^{\frac{-1}{2\psi_0^2}\left[\psi_1^2 + \frac{(\rho(x',y')\psi_1-\psi_2)^2}{1-\rho^2(x',y')}\right]}}{2\pi\psi_0^2\sqrt{1-\rho^2(x',y')}}\, d\psi_1 d\psi_2$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{e^{i(\psi_1-\psi_2)}\, e^{\frac{-1}{2\psi_0^2}\left[\psi_1^2 + \frac{(\rho(r')\psi_1-\psi_2)^2}{1-\rho^2(r')}\right]}}{2\pi\,\psi_0^2\sqrt{1-\rho^2(r')}}\, d\psi_1 d\psi_2 . \tag{2.4.14}$$

where $\rho(r') = \frac{<\phi_1\phi_2>}{\psi_0^2}$ and $r' = \sqrt{(x-\frac{1}{2}x'-(x+\frac{1}{2}x'))^2+(y-\frac{1}{2}y'-(y+\frac{1}{2}y'))^2}$. Integrating eq. (2.4.14) with respect to ψ_2, $<e^{i(\phi_1 - \phi_2)}>$ becomes

$$< e^{i(\phi_1-\phi_2)} > = \frac{e^{\psi_0^2(1-\rho(r'))/2}}{\psi_0\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-i(1-\rho(r'))\psi_1 - \psi_1^2/2\psi_0^2}\, d\psi_1$$

$$= e^{-\psi_0^2(1-\rho(r'))} \tag{2.4.15}$$

For large values of ψ_0^2, $< e^{i(\phi_1-\phi_2)} >$ is small as long as $(1- \rho(r'))$ is close to one. From eq. (2.4.6), the Taylor expansion of the correlation function about zero is

$$\rho(r') = 1 - \frac{r'^2}{2L^2} \tag{2.4.16}$$

By replacing $\rho(r')$ in eq. (2.4.15) with eq. (2.4.16), the phase factor in the integrand of $<f(\vec{x}_\perp, \vec{u}_\perp, z)>$, that is,

$$< f(\vec{x}_\perp,\vec{u}_\perp,z) > = \frac{k^2}{(2\pi)^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} d^2\vec{x}_\perp' e^{ik\vec{u}_\perp\cdot\vec{x}_\perp'}\epsilon_1(\vec{x}_\perp-\tfrac{1}{2}\vec{x}_\perp',z) \times \epsilon_1^*(\vec{x}_\perp+\tfrac{1}{2}\vec{x}_\perp',z)<e^{i(\phi_1-\phi_2)}> \tag{2.4.17}$$

becomes

$$< e^{i(\phi_1-\phi_2)} > = e^{-r'^2\psi_0^2/2L^2} \tag{2.4.18}$$

Following from this result, the distribution function at the lens plane is evaluated to be

$$< f(\vec{x}_\perp,\vec{u}_\perp,z_1) > = \frac{|E_0|^2k^2D^2a_0^2}{2\pi} e^{-2r^2/a_0^2} e^{\frac{k^2D^2a_0^2}{2}[(u_x+\frac{x}{f_L})^2+(u_y+\frac{y}{f_L})^2]} \tag{2.4.19}$$

where $D = \dfrac{1}{\sqrt{1+\psi_0^2a_0^2/L^2}}$

For zero values of ψ_0, D becomes one and perfect coherence of the beam results. On the other hand, for non-zero values of ψ_0, D is less than one, and phase coherence will be imperfect.

By substituting for kD by k_{eff}, the distribution function at the lens plane becomes

$$< f(\vec{x}_\perp,\vec{u}_\perp,z_1) > = \frac{|E_0|^2k_{eff}^2a_0^2}{2\pi} e^{\frac{-2r^2}{a_0^2}} e^{\frac{-k_{eff}^2a_0^2}{2}[(u_x+\frac{x}{f_L})^2+(u_y+\frac{y}{f_L})^2]} \tag{2.4.20}$$

And at any z-plane, the function becomes

$$<f(\vec{x}_\perp,\vec{u}_\perp,z)> = \frac{|E_0|^2k_{eff}^2a_0^2}{2\pi} e^{\frac{-2r^2}{a_0^2(z)}} e^{\frac{-k_{eff}^2a_0^2(z)}{2}[(u_x+\frac{x}{F(z)})^2+(u_y+\frac{y}{F(z)})^2]} \tag{2.4.21}$$

where

$$a^2(z) = a_0^2[(1-\frac{z-z_0}{f_L})^2 + \frac{4(z-z_0)^2}{k_{eff}^2a_0^4}] \tag{2.4.22}$$

and

$$F(z) = f_L\left[\frac{\left(1-\frac{z-z_0}{f_L}\right)^2 + \frac{4(z-z_0)^2}{k_{eff}^2 a_0^4}}{1-\frac{z-z_0}{f_L}\left(1+\frac{4f_L^2}{k_{eff}^2 a_0^4}\right)}\right] \tag{2.4.23}$$

The introduction of phase incoherence into the electric field amplitudes at the lens plane alters the size of the beam along the path of propagation as well as the focal distance. The incoherence factor D provides a means to adjust the beam size at the focus.

In conclusion, the phase space distribution function for a radiation field allows one to allocate the field energy into various rays of which positions and directions can be defined. Moreover, by using this function, the directions of the rays are distributed in such a way that a diffraction limited focal region can be reconstructed from the rays.

Chapter 3

Beam propagation in a refractive medium

In this chapter, the derivation of the ray equation is carried out and solutions for this equation are given for different profiles. These profiles are used to implement the density values calculated from Milroy's laser heated solenoid model.

3.1 Derivation of ray equation

In this section, the general equation for the trajectories traced by energy packets (photons) is derived.

Light propagation within a refractive medium can be described in terms of the motion of quasi-particles as long as the angular frequency, ω ,of the quasi-particle varies sufficiently slowly with respect to time and space[1,2,3]. This condition ensures that the wavepackets are localized in space. Equations of motion of the wavepacket are

$$\frac{d\vec{r}}{dt} = \dot{\vec{r}} = \frac{\partial\omega(\vec{k},\vec{r},t)}{\partial\vec{k}} \tag{3.1.1}$$

$$\frac{d\vec{k}}{dt} = \dot{\vec{k}} = \frac{\partial\omega(\vec{k},\vec{r},t)}{\partial\vec{r}} \tag{3.1.2}$$

Eq. (3.1.1) can be re-expressed as

$$\dot{\vec{r}} = (\frac{\partial\omega}{\partial k})\vec{\nabla}_k k = \vec{v}_g = v_g\vec{\Omega} \tag{3.1.3}$$

where

$$\vec{\Omega} = \vec{k}/k \tag{3.1.4}$$

and

$$v_g = \frac{\partial\omega}{\partial k} \tag{3.1.5}$$

As will be shown, eq. (3.1.1) can further be expressed in terms of the refractive index of the medium[4]. To start with, eq. (3.1.4) is differentiated with respect to time, giving

$$\dot{\vec{k}} = k\dot{\vec{\Omega}} + \dot{k}\vec{\Omega} \tag{3.1.6}$$

By combining with eq. (3.1.2), the equation becomes

$$k\dot{\vec{\Omega}} = -\vec{\nabla}_r\omega - \dot{k}\vec{\Omega} \tag{3.1.7}$$

By taking the dot products of eq. (3.1.2) with $\vec{k}$,

$$\vec{k}\cdot\dot{\vec{k}} = \vec{k}\cdot\frac{\partial\omega}{\partial\vec{r}}$$

$$\dot{k} = -\vec{\Omega}\cdot\vec{\nabla}_r\omega \tag{3.1.8}$$

then

$$k\dot{\vec{\Omega}} = -\vec{\nabla}_r\omega + (\vec{\Omega}\cdot\vec{\nabla}_r\omega)\vec{\Omega}$$

$$\dot{\vec{\Omega}} = -\frac{\vec{\nabla}_r\omega}{k} + \frac{(\vec{\Omega}\cdot\vec{\nabla}_r\omega)}{k}\vec{\Omega} \tag{3.1.9}$$

Let the refractive index be $\eta(\omega, r, t)$. The spatial gradient of ω is

$$\vec{\nabla}_r\omega = \vec{\nabla}_r\left(\frac{ck}{\eta}\right)$$

$$= ck\left[-\frac{\vec{\nabla}_r\eta}{\eta^2} - \frac{1}{\eta^2}\frac{\partial\eta}{\partial\omega}\frac{\partial\omega}{\partial r}\right]$$

$$= \frac{\dfrac{-ck\vec{\nabla}_r\eta}{\eta^2}}{\left(1+\dfrac{ck}{\eta^2}\dfrac{\partial\eta}{\partial\omega}\right)} \tag{3.1.10}$$

Also, the gradient of ω with respect to $\vec{k}$ is

$$\frac{\partial\omega}{\partial\vec{k}} = \frac{\partial}{\partial\vec{k}} \frac{ck}{\eta(\omega,\vec{r},t)}$$

$$= \frac{c}{\eta}\hat{k} - \frac{ck}{\eta^2}\frac{\partial\eta}{\partial\omega}\frac{\partial\omega}{\partial\vec{k}}$$

$$= \frac{\frac{c}{\eta}\hat{k}}{(1+\frac{ck}{\eta^2}\frac{\partial\eta}{\partial\omega})}$$

$$= \frac{\frac{c}{\eta}\hat{k}}{(1+\frac{\omega\partial\eta}{\eta\partial\omega})} \qquad (3.1.11)$$

where $\hat{k}$ is the unit vector of $\vec{k}$ and $\frac{ck}{\eta} = \omega$

But,

$$\left|\frac{\partial\omega}{\partial\vec{k}}\right| = |\vec{v}_g| = v_g$$

Substituting eq. (3.1.11) into eq. (3.1.10), we get

$$\vec{\nabla}_r\omega = -\frac{kv_g}{\eta}\vec{\nabla}_r\eta$$

$$\frac{\vec{\nabla}_r\omega}{k} = \frac{-v_g}{\eta}\vec{\nabla}_r\eta \qquad (3.1.12)$$

Combining eq. (3.1.12) and eq. (3.1.9), gives

$$\dot{\vec{\Omega}} = \frac{v_g}{\eta}\vec{\nabla}_r\eta - \frac{(v_g\vec{\Omega}\cdot\vec{\nabla}_r\eta)}{\eta}\vec{\Omega}$$

$$= \frac{v_g}{\eta}[\vec{\nabla}_r\eta - (\vec{\Omega}\cdot\vec{\nabla}_r\eta)\vec{\Omega}] \qquad (3.1.13)$$

In a plasma medium, the index of refraction is

$$\eta(\omega,\vec{r},t) = \sqrt{1 - \frac{\omega_{pe}^2(\vec{r},t)}{\omega^2}}$$

where $\omega_{pe}^2(\vec{r},t) = \frac{4\pi N(\vec{r},t)e^2}{m_e}$ and $N(\vec{r},t)$ is the electron density (cm^{-3}), m_e is the electron mass. Because of the extremely short time for light to traverse the plasma medium, the plasma density does not change fast enough to give a significant change in the refractive index of the medium. The plasma refractive index can be considered as constant within the time of traversing and regarded as dependent only on space.

By substituting for η into eq. (3.1.11), the group velocity is found to be

$$v_g = \eta(\vec{r},\omega)c \tag{3.1.14}$$

Eq. (3.1.13) thus becomes

$$\vec{\dot{\Omega}} = c\ [\ \vec{\nabla}_r\eta - (\vec{\Omega}\cdot\vec{\nabla}_r\eta\ )\vec{\Omega}\ ] \tag{3.1.15}$$

By multiplying both sides by η c, which is the velocity in the medium,

$$\eta c\dot{\vec{\Omega}} = \eta c^2\vec{\nabla}_r\eta - \eta c^2(\vec{\Omega}\cdot\vec{\nabla}_r\eta)\vec{\Omega}$$

and using the expression for group velocity,

$$\dot{\vec{r}} = \vec{v}_g = v_g\vec{\Omega} = \eta c\vec{\Omega}$$

then

$$v_g\dot{\vec{\Omega}} = \eta c^2\vec{\nabla}_r\eta - (\vec{v}_g\cdot\vec{\nabla}_r v_g)\vec{\Omega}$$

But,

$$\vec{v}_g\cdot\vec{\nabla}_r v_g = \frac{dv_g}{dt}$$

Hence,

$$v_g\dot{\vec{\Omega}} + \dot{v}_g\vec{\Omega} = \eta c^2\vec{\nabla}_r\eta$$

$$\frac{d(v_g\vec{\Omega})}{dt} = \frac{c^2}{2}\vec{\nabla}_r\eta^2$$

$$\frac{d^2\vec{r}}{dt^2} = \frac{c^2}{2}\vec{\nabla}_r\eta^2 \tag{3.1.16}$$

According to the above equation, as long as the refractive index is spatially known at the instant the rays traverse the medium, ray paths can traced.

3.2 Adoption of co-ordinate system

In this section, the choice of co-ordinate systems used for solving the ray equation is discussed.

The beam behaviour within a cylindrical plasma column is traced. Due to cylindrical symmetry of the column, cylindrical co-ordinate system is adopted. The ray equation can be expressed into its components as,

$$\frac{d^2r}{dt^2} = \frac{c^2}{2} \frac{\partial \eta^2(r,z)}{\partial r} + r\dot{\theta}^2 \tag{3.2.1}$$

$$\frac{d(r^2\dot{\theta})}{dt} = 0 \tag{3.2.2}$$

$$\frac{d^2z}{dt^2} = \frac{c^2}{2} \frac{\partial \eta^2(r,z)}{\partial z} \tag{3.2.3}$$

where $\dot{\theta}$ is the angular velocity of the ray. The refractive index is taken to be azimuthally independent. Eq.(3.2.2) indicates that the angular velocity component of the rays is a constant of motion due to the assumption of the azimuthal independence of the refractive index. Angular momentum of the ray is thus conserved.

In certain density profiles used later in this work, it will be more convenient to express the velocity components in the Cartesian co-ordinates for solving the ray equation. These components are given by

$$\frac{d^2x}{dt^2} = \frac{c^2}{2} \frac{\partial \eta^2(x,y,z)}{\partial x} \tag{3.2.4}$$

$$\frac{d^2y}{dt^2} = \frac{c^2}{2} \frac{\partial \eta^2(x,y,z)}{\partial y} \tag{3.2.5}$$

$$\frac{d^2z}{dt^2} = \frac{c^2}{2} \frac{\partial \eta^2(x,y,z)}{\partial z} \tag{3.2.6}$$

3.3 Choice of density profile

In this section, the ray equation is solved for different refractive indices which are determined from the density of the plasma confined in the solenoid. The behaviour of the ray is discussed accordingly.

In solving the ray equation within a plasma region, it is necessary to know the spatial variation of the refractive index in that region. The refractive index for a plasma is given as

$$\eta(\vec{r}) = \sqrt{1-\frac{\omega_{pe}^2}{\omega^2}} = \sqrt{1-\frac{N(\vec{r})}{N_c}} \qquad (3.3.1)$$

where

$$\omega_{pe}^2(\vec{r}) = \frac{4\pi N(\vec{r})e^2}{m_e}$$

$N(\vec{r})$ is the plasma density at a distance $\vec{r}$ from the origin ;N_c is the critical density, $\frac{m_e\omega^2}{4\pi e^2}$ ω is the angular frequency of the laser. The refractive index at any point is thus determined directly from the plasma density at that point.

In McMullin and Milroy's simulation work,the plasma region is divided into meshes by means of a two-dimensional grid structure. The characteristics of the plasma such as energy, temperature and density are evaluated at the centre of the grid cells. These physical quantities are hence known only at discrete positions. In order to trace the beam trajectory within the plasma simulated by the MHD code, a knowledge of the density variation between any two adjacent cells is essential. Moreover, if the density variation is known to be a continuous function of position, the ray equation can be solved to give a continuous solution for the path between two adjacent grid cells.

Since plasma densities are only evaluated at the centres of the grid cells, appropriate continuous functions have to be used to describe the density variation between adjacent ones. The choice of such functions is deduced from the physical behaviour of the plasma when heated by means of a laser beam. These functions give a profile description of the plasma density in the corresponding region.

Density profiles which are functions of the square or inverse square of radial distance are used since they can give a suitable description for the density hollow created by a laser beam within the plasma column. Such a choice also enables the ray equation to be solved analytically. The density profiles are assumed to be azimuthally independent as a result of cylindrical symmetry of the plasma column.

Four different kinds of density profiles are adopted in accordance with the radial density changes at adjacent grid cells. They are as follows:

(1) $$\frac{dN(r)}{dr} > 0, \quad \frac{d^2N(r)}{dr^2} > 0, \quad N(r) = N_0\left(1 + \frac{r^2}{a_0^2}\right) \qquad (3.3.2)$$

(2) $$\frac{dN(r)}{dr} > 0, \quad \frac{d^2N(r)}{dr^2} < 0, \quad N(r) = N_1\left(1 - \frac{a_1^2}{r^2}\right) \qquad (3.3.3)$$

(3) $$\frac{dN(r)}{dr} < 0, \quad \frac{d^2N(r)}{dr^2} > 0, \quad N(r) = N_2\left(1 - \frac{r^2}{a_2^2}\right) \qquad (3.3.4)$$

(4) $$\frac{dN(r)}{dr} < 0, \quad \frac{d^2N(r)}{dr^2} < 0, \quad N(r) = N_3\left(1 + \frac{a_3^2}{r^2}\right) \qquad (3.3.5)$$

These profiles are fitted into corresponding regions wherever the density differences satisfy the conditions for $\frac{dN}{dr}$ and $\frac{d^2N}{dr^2}$.

The suggested profiles are used to approximate various regions of the laser heated plasma column (fig. 3.1). The density distribution across the column can hence be approximated by segments of continuous functions. The core part of the column is described by a parabolic well corresponding to a density hollow created by laser heating (profile #1 in fig. 3.1) profile. Further from the core, the plasma density reaches a maximum due to the accumulation of plasma particles resulting from radial expansion of the plasma. The region between this density maximum and the parabolic well is simulated with profile #2 in fig. 3.1. Beyond this point, the plasma density decreases gradually. This region is approximated by profile #3 in fig. 3.1. On getting closer to the wall of the solenoid, profile #4 in fig. 3.1 is used. The density distribution across the column is thus described by segments of continuous functions.

With the refractive index given as in eq. (3.3.1), (r becomes the radial distance only, because of azimuthal and axial independence), the ray equation, as derived from eq.

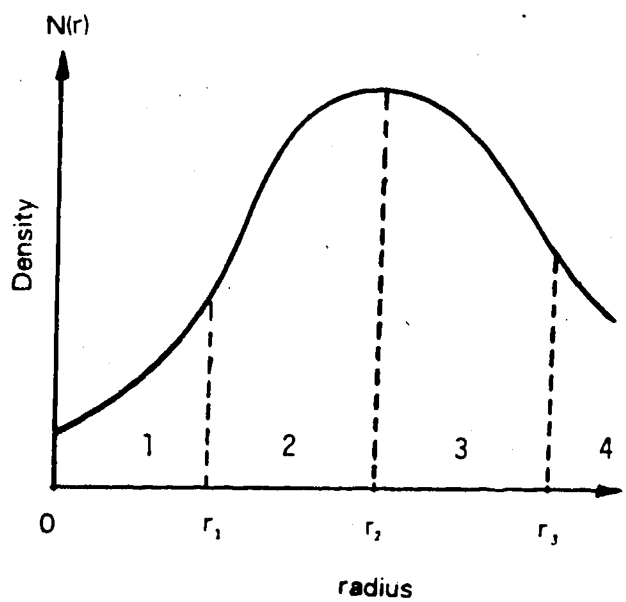


Figure 3.1 Radial density profile of the column

(3.2.1), eq. (3.2.2) and eq. (3.2.3), becomes

$$\frac{d^2r}{dt^2} = \frac{c^2}{2N_c}\frac{dN(r)}{dr} + r\dot{\theta}^2 \tag{3.3.6}$$

$$\frac{d(r^2\dot{\theta})}{dt} = 0 \tag{3.3.7}$$

$$\frac{d^2z}{dt^2} = 0 \tag{3.3.8}$$

Eq. (3.3.7) and eq. (3.3.8) can be solved, with given initial ray conditions, for the angular velocity and the axial position. In what follows, the solutions of the differential equation for the radial position in different profiles are discussed as follow:

(3.3.1) Parabolic radial profile

This profile is applied to the regions between two adjacent grid cells where $\frac{dN(r)}{dr}>0$ and $\frac{d^2N(r)}{dr^2}>0$(region 1 of fig. 3.1). The parabolic profile is

$$N(r) = N_0(1 + \frac{r^2}{a_0^2}) = N_0(1 + \frac{x^2+y^2}{a_0^2}) \tag{3.3.1.1}$$

where N_0, a_0 are parameters determined from the boundary conditions of N(r);and r is the radial distance. This profile applies frequently in the region near the axis of the plasma column. In Cartesian co-ordinates the equations of motion are

$$\frac{d^2x}{dt^2} = \frac{c^2N_0x}{a_0^2N_c} = -\Omega^2x \tag{3.3.1.2}$$

$$\frac{d^2y}{dt^2} = \frac{c^2N_0y}{a_0^2N_c} = -\Omega^2y \tag{3.3.1.3}$$

$$\frac{d^2z}{dt^2} = 0 \tag{3.3.1.4}$$

where

$$\Omega^2 = \frac{c^2N_0}{a_0^2N_c}$$

The solution for these equations is that of a simple harmonic motion. With initial conditions,

$t=0, x=x_0, y=y_0, \dot{x}=\dot{x}_0, \dot{y}=\dot{y}_0.$

$$x = x_0\cos(\Omega t) + \frac{\dot{x}_0}{\Omega}\sin(\Omega t) \tag{3.3.1.5}$$

$$y = y_0\cos(\Omega t) + \frac{\dot{y}_0}{\Omega}\sin(\Omega t) \tag{3.3.1.6}$$

$$\dot{x} = -\Omega x_0 \sin(\Omega t) + \dot{x}_0 \cos(\Omega t) \tag{3.3.1.7}$$

$$\dot{y} = -\Omega y_0 \sin(\Omega t) + \dot{y}_0 \cos(\Omega t) \tag{3.3.1.8}$$

$$r^2 = \frac{r_0^2}{2} + \frac{v_{0r}^2}{2\Omega^2} + B\sin(2\Omega t + \phi) \tag{3.3.1.9}$$

where

$$\phi = \tan^{-1}\left[\frac{\left(\frac{r_0^2}{2} - \frac{v_{0r}^2}{2\Omega^2}\right)}{\left(\frac{x_0\dot{x}_0+y_0\dot{y}_0}{\Omega}\right)}\right]$$

$$B^2 = \left(\frac{r_0^2}{2} - \frac{v_{0r}^2}{2\Omega^2}\right)^2 + \left(\frac{x_0\dot{x}_0+y_0\dot{y}_0}{\Omega}\right)^2$$

$$v_{0r}^2 = \dot{x}_0^2 + \dot{y}_0^2$$

$$r_0^2 = x_0^2 + y_0^2$$

The sinusoidal variation of the ray path with time indicates that the ray is trapped within the medium. A real solution for r^2 is guaranteed when $B<r_0^2/2+v_{0r}^2/2\Omega^2$. The last condition will depend on the initial conditions of the ray. A negative solution for r^2 indicates that the ray is not launched at the right position or direction and that the ray cannot go through the medium. From eq. (3.3.1.9), the maximum and minimum radial distances are given by

$$r^2 = \frac{r_0^2}{2} + \frac{v_{0r}^2}{2\Omega^2} \pm B \quad \text{(+ for maximum; - for minimum)}$$

(3.3.2) Inverse parabolic increasing profile

This profile is used wherever the density difference between any two adjacent grid cells satisfies the conditions $\frac{dN(r)}{dr} > 0$ and $\frac{d^2N(r)}{dr^2} < 0$. The region defined within radius r_1, r_2 (region 2 of fig. 3.1) can be approximated by this profile. The profile is

$$N(r) = N_1(1 - \frac{a_1^2}{r^2}) \tag{3.3.2.1}$$

with $r \neq 0$. N_1, a_1 are constants determined from the boundary conditions of N(r) in the region under consideration. On substituting for N(r) from eq. (3.3.2.1) into the ray equation, the solution expressed in cylindrical co-ordinates, is

$$\frac{d^2r}{dt^2} = -\frac{c^2N_1a_1^2}{N_c r^3} + \frac{p^2}{r^3}$$

$$= (p^2 - \frac{c^2N_1a_1^2}{N_c})\frac{1}{r^3} \tag{3.3.2.2}$$

$$\frac{d(r^2\dot{\theta})}{dt} = 0 \tag{3.3.2.3}$$

$$\frac{d^2z}{dt^2} = 0 \tag{3.3.2.4}$$

With the initial conditions, $t=0$, $r=r_0$, $\dot{\theta}=\dot{\theta}_0$, $p=r_0^2\dot{\theta}_0$, $v_r=v_{0r}$, the radial velocity is found to be

$$\dot{r}^2 = (\frac{dr}{dt})^2 = v_{0r}^2 + (p^2 - \frac{c^2a_1^2N_1}{N_c})(\frac{1}{r_0^2} - \frac{1}{r^2}) \tag{3.3.2.5}$$

By integrating this equation with respect to time, the expression for the radial position can be obtained. Cases for outward radial acceleration and inward radial acceleration are considered as follows:

(i) For the case of outward radial acceleration, the quantity

$p^2 - \frac{c^2a_1^2N_1}{N_c}$ is positive. The corresponding solution for the equation is

$$t = \frac{r_0}{r_0^2v_{0r}^2+(p - \frac{c^2a_1^2N_1}{N_c})}[\pm\sqrt{r^2(r_0^2v_{0r}^2+(p^2-\frac{c^2a_1^2N_1}{N_c}))-r_0(p^2-\frac{c^2a_1^2N_1}{N_c})} \frac{r_0^2v_{0r}^2}{r_0^2v_{0r}^2+(p^2-\frac{c^2a_1^2N_1}{N_c})} \quad (3.3.2.6)$$

for $v_{0r}>0$, t takes the positive root; for $v_{0r}<0$, t takes the negative root. The radial distance is given by

$$r = \frac{1}{\sqrt{(r_0^2v_{0r}^2+p^2-\frac{c^2a_1^2N_1}{N_c})}}[(p^2-\frac{c^2a_1^2N_1}{N_c})r_0^2+\frac{1}{r_0^2}(r_0^2v_{0r}^2+p^2-\frac{c^2a_1^2N_1}{N_c})^2$$

$$\times\ (t\pm\frac{r_0^3v_{0r}}{r_0^2v_{0r}^2+p^2-\frac{c^2a_1^2N_1}{N_c}})^2]^{\frac{1}{2}}$$

With an initial inward radial velocity, the ray will move towards the axis of propagation until it reaches a minimum radial position where it is refracted away from the axis. From eq. (3.3.2.5), one can deduce that the minimum radial position of the ray is

$$r_{min} = \sqrt{[\frac{r_0^2(p^2-\frac{c^2a_1^2N_1}{N_c})}{(r_0^2v_{0r}^2+p^2-\frac{c^2a_1^2N_1}{N_c})^{\frac{1}{2}}}]} \quad (3.3.2.7)$$

(ii) For the case of inward radial acceleration, that is $p^2-\frac{c^2a_1^2N_1}{Nc}< 0$,

the square of radial position will be given by

$$r^2 = \frac{1}{r_0^2v_{0r}^2-|p^2-\frac{c^2a_1^2N_1}{N_c}|}[\frac{1}{r_0^2}(r_0^2v_{0r}^2-|p^2-\frac{c^2a_1^2N_1}{N_c}|)^2$$

$$\times\ (t \pm \frac{r_0^3v_{0r}}{r_0^2v_{0r}^2-|p^2-\frac{c^2a_1^2N_1}{N_c}|})^2-r_0^2|p^2-\frac{c^2a_1^2N_1}{N_c}|] \quad (3.3.2.8)$$

When $r_0^2 v_{0r}^2 - |p^2 - \frac{c^2 a_1^2 N_1}{N_c}| > 0$ and v_{0r} is < 0, the solution of r is imaginary. This implies that rays of which the initial locations and directions satisfy these conditions, cannot penetrate this region.

(3.3.3) Parabolic decreasing radial profile

This approximation is used in regions where the spatial density variation follows the relations $dN(r)/dr<0$ and $d^2N(r)/dr^2>0$ (region 3 of fig. 3.1). A possible region in the density well is beyond where the plasma density reaches its maximum (region bounded by the radii r_2 and r_3 in fig. 3.1). The density profile is expressed as

$$N(r) = N_2(1 - \frac{r^2}{a_2^2}) \tag{3.3.3.1}$$

Solution for the ray path expressed in Cartesian co-ordinates, is

$$x = \frac{\dot{x}_0}{\Omega} \sinh(\Omega t) + x_0 \cosh(\Omega t) \tag{3.3.3.2}$$

$$y = \frac{\dot{y}_0}{\Omega} \sinh(\Omega t) + y_0 \cosh(\Omega t) \tag{3.3.3.3}$$

$$\dot{x} = \dot{x}_0 \cosh(\Omega t) + \Omega x_0 \sinh(\Omega t) \tag{3.3.3.4}$$

$$\dot{y} = \dot{y}_0 \cosh(\Omega t) + \Omega y_0 \sinh(\Omega t) \tag{3.3.3.5}$$

$$r^2 = (\frac{r_0^2}{2} - \frac{v_{0r}^2}{2\Omega^2}) + (\frac{r_0^2}{2} + \frac{v_{0r}^2}{2\Omega^2})\cosh(2\Omega t) + (\frac{x_0\dot{x}_0 + y_0\dot{y}_0}{\Omega})\sinh(2\Omega t) \tag{3.3.3.6}$$

where x_0, y_0 are the initial positions of the ray, $r_0^2 = x_0^2 + y_0^2$, $v^2 = \dot{x}_0^2 + \dot{y}_0^2$.

$\Omega^2 = \dfrac{c^2 N_2}{N_c a_2^2}$, $\dot{x}_0, \dot{y}_0$ are the initial velocities.

Depending on the initial conditions of the ray, the ray will traverse in this region radially inward or radially outward. If all terms in eq (3.3.3.6) are positive, the radial distance will increase directly with t. The ray will leave the region gradually and will not be trapped. r^2 will become negative according to the initial ray conditions. A negative value for r^2 indicates that the ray may launch into the plasma region such that it is reflected off. When the ray goes radially inward, the minimum radial distance that the ray will approach is,

$$r_{min}^2 = \left(\frac{r_0^2}{2} - \frac{v_{0r}^2}{2\Omega^2} \right) + \sqrt{\left(\frac{r_0^2}{2} + \frac{v_{0r}^2}{2\Omega^2} \right)^2 - \left(\frac{x_0\dot{x}_0 + y_0\dot{y}_0}{\Omega} \right)^2} \qquad (3.3.3.7)$$

which can be seen to be dependent on the initial position and velocity of the ray.

(3.3.4) Non-parabolic decreasing radial profile

This profile fits into the region where the relations $\dfrac{dN(r)}{dr} < 0$ and $\dfrac{d^2N(r)}{dr^2} < 0$ hold(region 4 of fig. 3.1). The density profile in the region is given as

$$N(r) = N_3\left(1 + \frac{a_3^2}{r^2} \right) \qquad (3.3.4.1)$$

with $r \neq 0$. By using cylindrical co-ordinates and initial conditions v_{0r}, r_0, the radial location of the ray trajectory is given by

$$r = \frac{1}{\left(r_0^2 v_{0r}^2 + p^2 + \frac{c^2 a_3^2 N_3}{N_c} \right)^{\frac{1}{2}}} \left[r_0^2\left(p^2 + \frac{c^2 a_3^2 N_3}{N_c} \right) + \frac{1}{r_0^2}\left(r_0^2 v_{0r}^2 + p^2 + \frac{c^2 a_3^2 N_3}{N_c} \right)^2 \right.$$

$$\left. \times \left(t \pm \frac{r_0^3 v_{0r}}{r_0^2 v_{0r}^2 + p^2 + \frac{c^2 a_3^2 N_3}{N_c}} \right)^2 \right]^{\frac{1}{2}} \qquad (3.3.4.2)$$

Through comparing the two expressions for r given by eq. (3.3.4.2) and eq. (3.3.2.6), it can be noted that only the term $(p^2+c^2a_3^2N_3/N_C)$ is changed. Similar results can thus be concluded for the inward radial velocity case. The ray will approach its minimum radial distance, according to the relation,

$$r_{min}^2 = \frac{r_0^2\left(p^2+\frac{c^2a_3^2N_3}{N_c}\right)}{r_0^2v_{0r}^2+p^2+\frac{c^2a_3^2N_3}{N_c}} \qquad (3.3.4.3)$$

The adoption of the four profiles for describing the plasma density in the column is initiated from the density well shown in fig. 3.1. Regions within the plasma column are matched to the corresponding density profiles according to the sign of the density gradient and the derivative of the density gradient within that region. The density gradient conditions $\frac{dN(r)}{dr}>0$ or <0 and the derivative of the density gradient $\frac{d^2N(r)}{dr^2}>0$ or <0 determine the type of density profiles to be used in computing the ray trajectories. In previous work[5,6,7], only parabolic profiles were assumed within the column. The corresponding ray path is sinusoidal as was discussed in section (3.1). However, in McMullin and Milroy's MHD code, the plasma density profile is arbitrary and determined only through a self-consistent set of fluid equations. Both parabolic and non-parabolic profiles are used for simulating the density distribution. Such choice eliminates the use of numerical methods to solve the ray equation. However, corresponding parameters for various profiles are needed to be evaluated for each set of density values calculated from the MHD code. The axial density variation is assumed to be slow so that there is no significant change over one axial grid distance. This assumption is valid as long as there is no abrupt axial density change encountered, otherwise axial density variation has to be included.

Chapter 4

Laser power absorption and ponderomotive forces

The deposition of laser energy within the plasma medium is discussed in this chapter. The ponderomotive force due to high laser intensity is considered as well. In section 4.1, the distribution of power among rays is discussed. and in section 4.2 an account of the absorption mechanism involved in the energy transfer process is given. Moreover, the ponderomotive force due to inhomogeneous laser intensity distribution is considered in section 4.3.

4.1 Power carried by individual rays

In this section, methods of distributing the radiation power among the rays are briefly discussed.

In a cylindrical co-ordinate system, the radial power of a Gaussian beam at a radial distance r, can be obtained by integrating the intensity distribution I(r) over the cross sectional area with radius r. From eq. (2.2.2), the amplitude of the Gaussian beam across a transverse cross section at the lens plane is

$$|\varepsilon(r,\ z_0)| = \sqrt{\frac{2}{\pi}}\,\frac{1}{\omega(z_0)}\,e^{\frac{-r^2}{\omega^2(z_0)}} \tag{4.1.1}$$

The intensity of the beam transmitted through an area of radius r is

$$\begin{aligned} I(r) &= 2\pi\int_0^r |\varepsilon(r',\ z_0)|^2 r'dr' I_0 \\ &= 2\pi\int_0^r I_0\,\frac{2}{\pi}\,\frac{1}{\omega^2(z_0)}\,e^{\frac{-2r'^2}{\omega^2(z_0)}}\,r'dr' \\ &= I_0\left(1 - e^{\frac{-2r^2}{\omega^2(z_0)}}\right) \end{aligned} \tag{4.1.2}$$

where I_0 is the total intensity.

The power transmitted through this area is

$$P(r) = \frac{cI(r)}{8\pi} = \frac{cI_0}{8\pi}\left(1 - e^{\frac{-2r^2}{\omega^2(z_0)}}\right) \tag{4.1.3}$$

This power can be distributed among the rays simulating the beam by using the distribution function derived for an incoherent Gaussian beam (eq. 2.4.19), namely,

$$< f(x,y,u_x,y_y,z) > = \frac{|E_0|^2k^2D^2a_0^2}{2\pi} e^{-\frac{2(x^2 \;\; y^2)}{a_0^2}} \times e^{\frac{-k^2D^2a_0^2}{2}[(u_x+\frac{x}{F(z)})^2+(u_y+\frac{y}{F(z)})^2]} \tag{4.1.4}$$

The energy density associated with a ray at a location (x,y) and a direction (u_x, u_y) is evaluated from eq. (4.1.4) and is given by $\int<f>du_\perp$. Power transmitted along the axial direction is given by $\frac{c}{8\pi}\int<f>du_\perp$, where c is the speed of light. The power carried by a ray can be determined directly once its position and direction are known. The set of rays used for describing the beam can be determined in two ways. Firstly, the positions and directions of the rays can be preset so that the ray density is constant across the beam. The associated power carried by each ray is then evaluated from eq. (4.1.3) in terms of its predetermined position and direction. Alternatively, the rays can be assumed to carry equal power. The power density within a certain beam area is determined by counting the number of rays within the area. Locations and directions of the the rays can be determined according to a random Gaussian distribution which will be discussed in section 5.1.

4.2 Derivation of the absorption coefficient along a ray

In this section, the coefficient of absorption in a plasma along unit length of the rays is derived. The power absorbed along a ray is then evaluated.

For the case of laser heated solenoid, the plasma absorbs laser energy through the process of inverse Bremsstrahlung. This process dominates over other laser heating mechanisms(see footnote) because laser intensity is in the order of less than $10^{11}-10^{12}$ W/cm^2 which is not strong enough to initiate other heating processes.

From Johnston and Dawson[1], the absorption coefficient per unit length for this process for an incident monochromatic beam wave with a wavelength of 10.6μm, is

---

Anomalous ion and electron heating resulting from parametric excitation of plasma instabilities

$$K_a = \frac{8.67\times10^{-30}\, Ze^2 n\lambda^2 \ln\Lambda}{T_e^{\frac{3}{2}}\left(1 - \frac{\omega_{pe}^2}{\omega^2}\right)} \quad cm^{-1} \tag{4.2.1}$$

where

$$\Lambda = \min\left[\frac{2.19\times10^3 \times T_e^{\frac{3}{2}}\lambda(cm)}{Z},\ 1.14\lambda_{cm} T_e\right]$$

The absorption coefficients along ray paths can be found by considering absorption on small segments of the trajectory. This is explained as follows. For a ray segment of length ds, the associated absorption coefficient within a grid cell(see footnote) is

$$dK = K_a ds \tag{4.2.2}$$

Also ds is the length of the trajectory travelled by a photon in a time dt, at a velocity n(r)c (n(r) is the refractive index of the medium). The equation can thus be written as

$$dK = K_a n(r(t))c\ dt \tag{4.2.3}$$

where r(t) is the radial position of the photon at time t. The power transferred to the medium within this segment ds is

$$dP(r,t) = -K_a n(r(t))cP(r,t)dt \tag{4.2.4}$$

Since the time interval(dt) for the beam to propagate along a distance ds within the plasma column is relatively short (10^{-9} sec.) compared to the pulse width of the laser(in the order of 10^{-6} sec.), the beam power from the laser can be regarded as constant over the time (dt) for traversing the distance ds. The amount of power left, P_f, after a time dt is

---

The plasma column is divided into tiny volumes by a mesh. Each volume is termed as a grid cell.

$$\int_{P_i}^{P_f} \frac{dP(r,t_0)}{P(r,t_0)} = -\int_0^{dt} K_a n(r(t'))c\,dt' \quad (4.2.5)$$

$$P_f(r_0+dr,t+dt) = P_i(r_0,t_0)e^{-\int_0^{dt} K_a n(r(t'))c\,dt'} \quad (4.2.6)$$

where $P_i(r_0,t_0)$ is the beam power at time t_0 and location r_0. Within the time interval dt, the input beam power is assumed to be constant with time and has the value $P_i(r_0, t_0)$. The amount of absorbed power is

$$P_{abs} = P_i(r_0,t_0) - P_f(r_0+dr,t_0+dt)$$

$$= P_i(r_0,t_0)[\,1 - e^{-\int_0^{dt} K_a n(r(t'))c\,dt'}\,] \quad (4.2.7)$$

The beam loses ΔP_{abs} watts for each traverse of the distance ds.

4.3 Calculation of ponderomotive forces

In this section, the ponderomotive force is derived according to Chen's analysis[2].

As a laser beam is focused onto a plasma medium, the intensity at the focal region is extremely high. The electromagnetic field which is proportional to intensity rises significantly. The consequent effect is that the field imposes a strong Lorentz force on the particles leading to a charge separation which bundles up electrons and ions into discrete regions. This force acting on an electron, with mass m_e, located at r is

$$m_e\left(\frac{d\vec{v}}{dt}\right) = -e[\,\vec{E}(\vec{r}) + \frac{1}{c}\vec{v} \times B(\vec{r})\,] \quad (4.3.1)$$

where $\vec{E}(\vec{r})$ is the electric field of the radiation ; $\vec{v}$ is the velocity of the electrons. With an incident electric field given as

$$\vec{E}(\vec{r},t) = \vec{E}_s(\vec{r})\cos(\omega_0 t) \quad (4.3.2)$$

where ω_0 is the angular frequency of the incident wave. $\vec{E}_s(\vec{r})$ is the electric field with spatial dependence only. The corresponding magnetic field is found from Maxwell's equation

$$\frac{1}{c}\frac{\partial\vec{B}}{\partial t} = -\vec{\nabla} \times \vec{E}$$

$$\vec{B}(\vec{r}) = \frac{c}{\omega_0} \vec{\nabla} \times \vec{E}_s(\vec{r})\sin(\omega_0 t) \tag{4.3.3}$$

To the first order, the force on an electron at location $\vec{r}$ can be evaluated by taking $\vec{E}$ at the initial position $\vec{r}$, and neglecting the term $\vec{v} \times \vec{B}$ since it is smaller than $\vec{E}$ by a factor of v/c. The force is

$$m_e\left(\frac{d\vec{v}}{dt}\right) = -e\vec{E}(\vec{r}_0) \tag{4.3.4}$$

On integrating, and using the initial conditions that the electron has a zero velocity and locates at the maximum amplitude of oscillation, the first order velocity perturbation is

$$\vec{v}^{(1)} = \frac{-e}{m_e\omega_0} \vec{E}_s(\vec{r})\sin(\omega_0 t) \tag{4.3.5}$$

and upon further integration, the first order displacement perturbation of the electron from the position r_0 is,

$$\delta\vec{r}^{(1)} = \int \vec{v}^{(1)}\, dt = \frac{e}{m_e\omega_0^2} \vec{E}_s(\vec{r})\cos(\omega_0 t) \tag{4.3.5a}$$

By using the velocity, and a first order expansion of $\vec{E}(\vec{r})$ in eq. (4.3.1) about the electron initial position $\vec{r}_0$, a second order approximation of the force is found to be

$$m_e\left(\frac{d\vec{v}}{dt}\right)^{(2)} = -e[(\delta\vec{r} \cdot \vec{\nabla})\vec{E}(\vec{r}_0) + \frac{1}{c}\vec{v}^{(1)} \times \vec{B}(\vec{r}_0)] \tag{4.3.6}$$

where $\vec{v}^2$ is a second order correction term to velocity $\vec{v}$. By taking the time average of eq. (4.3.6), and using eq. (4.3.3), (4.3.5),and (4.3.5a),the average non-linear force on an electron is thus

$$\vec{f}_{NL} = m_e\left\langle\frac{d\vec{v}^{(2)}}{dt}\right\rangle = \frac{e^2}{m_e\omega_0^2}\,\frac{1}{2}\,[(\vec{E}_s \cdot \vec{\nabla})\vec{E}_s + \vec{E}_s \times \vec{\nabla} \times \vec{E}_s] \tag{4.3.7}$$

The $\vec{E}_s \times \vec{\nabla} \times \vec{E}_s$ force component (which is equal to $\vec{E}_s \times \vec{k} \times \vec{E}_s$ upon Fourier transform) acts along the direction of propagation and causes both electrons and ions to move along the direction of propagation, while the $(\vec{E}_s \cdot \vec{\nabla})\vec{E}_s$ denotes the force component acting along the direction of the electric field vector. This force pushes the plasma to bundle up in a direction perpendicular to the direction of propagation. As a result, the formation of a low density region along the column is enhanced.(see footnote) This causes a change in the refractive index which will cause the beam to be focused and defocused as it propagates within the medium. Using the vector identity,

$$\frac{1}{2}\vec{E}_s = (\vec{E}_s \cdot \vec{\nabla})\vec{E}_s + \vec{E}_s \times \vec{\nabla} \times \vec{E}_s$$

the ponderomotive force per unit volume is given by Chen,

$$\vec{F}_{NL} = \frac{-\omega_{pe}^2}{\omega_0^2}\vec{\nabla}\left(\frac{E_s^2(\vec{r})}{16}\right) \tag{4.3.8}$$

where $\omega_{pe}^2 = \frac{4\pi N_0 e^2}{m_e}$; m_e is the electron mass, N_0 is the plasma density. However, the intensity of the radiation field is also given by

$$I = \frac{\eta c |E_A(\vec{r}_0)|^2}{8\pi} \tag{4.3.9}$$

where η is the index of refraction of the medium.
The ponderomotive force thus becomes

$$\vec{F}_{NL} = \frac{\omega_{pe}^2}{2\omega_0^2}\vec{\nabla}\left(\frac{I}{nc}\right) \tag{4.3.10}$$

Due to cylindrical symmetry, there is no azimuthal intensity gradient for the beam within the plasma. So only radial and axial components of the ponderomotive force have to be considered. The relevant components, in cylindrical co-ordinate system, are

$$(\vec{F}_{NL})_r = \frac{\omega_{pe}^2}{2\omega_0^2}\frac{\partial}{\partial r}\left(\frac{I}{nc}\right)\hat{r} \tag{4.3.11}$$

Plasma heating is the major factor for this formation.

$$(\vec{F}_{NL})_z = \frac{\omega_{pe}^2}{2\omega_0^2} \frac{\partial}{\partial z}\left(\frac{I}{nc}\right)\hat{z} \qquad (4.3.12)$$

where $\hat{r}$, $\hat{z}$ are unit vectors in radial and axial direction respectively. These force components can be known once the spatial variation in the beam intensity is known.

Chapter 5

Computational method

Numerical treatment of the analytical results in chapters three and four are dealt with in this chapter. Program packages are designed for computing ray distributions, density gradients, ray paths within plasma medium, power absorption along the paths and ponderomotive forces. These packages are designed to replace the laser profile routine used in the magnetic flux shell code.

5.1 Ray distribution package

In this section, the distribution of locations and directions of the rays is deduced from the distribution function for a coherent gaussian beam. Ray locations and directions are distributed according to a Gaussian distribution function.

Beam propagation is simulated in terms of bundles of rays. From the discussion in section (4.1), rays can be either chosen to carry power which varies according to a Gaussian distribution or to carry equal power. For rays carrying different power,the locations and directions must be predetermined. However, for rays carrying equal power,the locations and directions are so chosen that the number of rays varies according to a Gaussian distribution function. This scheme is explained as follows.

The initial power(P_{avg}) for each ray is just the beam power(P_0) at the lens plane at time t averaged over the total number of rays (N_0).That is,

$$P_0 = P_{avg}N_0 \tag{5.1.1}$$

From Tappert's phase-space distribution function for an incoherent beam at the lens plane,namely,

$$f(x,y,u_x,u_y,z_1) = \frac{|E_0|^2k^2a_0^2D^2}{2\pi} e^{\frac{-2(x^2+y^2)}{a_0^2}} e^{\frac{-k^2a_0^2D^2}{2}[(u_x+\frac{x}{f_L})^2+(u_y+\frac{y}{f_L})^2]} \tag{5.1.2}$$

the beam power density evaluated at the point (x,y) is

$$P(x,y) = \frac{c}{8\pi}\int_{-1}^{1}\int_{-1}^{1} \frac{|E_0|^2k^2D^2a_0^2}{2\pi}\, e^{\frac{-2(x^2+y^2)}{a_0^2}}\, e^{\frac{-k^2a_0^2D^2}{2}[(u_x+\frac{x}{f_L})^2+(u_y+\frac{y}{f_L})^2]}\, du_x du_y$$

(5.1.3)

If there are N(x,y)dxdy rays passing through an infinitesimal area dxdy at location (x,y,z_1), the beam power can be expressed as

$$P(x,y)dxdy = P_{avg}N(x,y)dxdy$$

(5.1.4)

By comparing eqs.(5.1.3) and (5.1.4), the number density of rays passing through the point can be seen to be given by

$$N(x,y) = \frac{c|E_0|^2k^2D^2a_0^2}{8\pi P_{avg}}\, e^{\frac{-2(x^2+y^2)}{a_0^2}} \int_{-1}^{1}\int_{-1}^{1} e^{\frac{-k^2a_0^2D^2}{2}[(u_x+\frac{x}{f_L})^2+(u_y+\frac{y}{f_L})^2]}\, du_x du_y dxdy$$

$$= \frac{c|E_0|^2k^2D^2a_0^2}{8\pi P_{avg}}\, e^{\frac{-2(x^2+y^2)}{a_0^2}} \left[\int_{-1}^{1}\frac{1}{\sqrt{2\pi}}\, e^{\frac{-k^2D^2a_0^2}{2}(u_x+\frac{x}{f_L})^2} du_x\right.$$

$$\left.\times \int_{-1}^{1}\frac{1}{\sqrt{2\pi}}\, e^{\frac{-k^2D^2a_0^2}{2}(u_y+\frac{y}{f_L})^2} du_y\right]\, dx\, dy$$

(5.1.5)

The expression shows that the total number density of rays is distributed gaussianly with respect to locations and directions.

In actual simulation of the laser beam, a set of rays is chosen such that they are gaussianly distributed over the lens plane. Moreover, the directions of the rays are so selected that the number of rays will be gaussianly distributed about the direction pointing towards the focus.

Such choice is achieved by using a normal random deviate generator. The generator will generate numbers which are normally distributed. A set of such numbers is designated to each of the locations x, y and directions u_x, u_y. For the x, y locations, the generated numbers are scaled according to the laser beam width to give actual spatial ray locations. For the directions, u_x, u_y, the generated numbers n_x, n_y are transformed to the real ray directions through the relations

$$n_x = (u_x + \frac{x}{f_L}) k a_0 D \tag{5.1.6}$$

$$n_y = (u_y + \frac{y}{f_L}) k a_0 D \tag{5.1.7}$$

These relations are deduced from the cumulative normal distribution function used in the numb[illegible] namely, (fig. 5.1)

$$P_R(t) = \int^{t} \sqrt{\frac{1}{2\pi}}\, e^{\frac{-T^2}{2}}\, dT \tag{5.1.8}$$

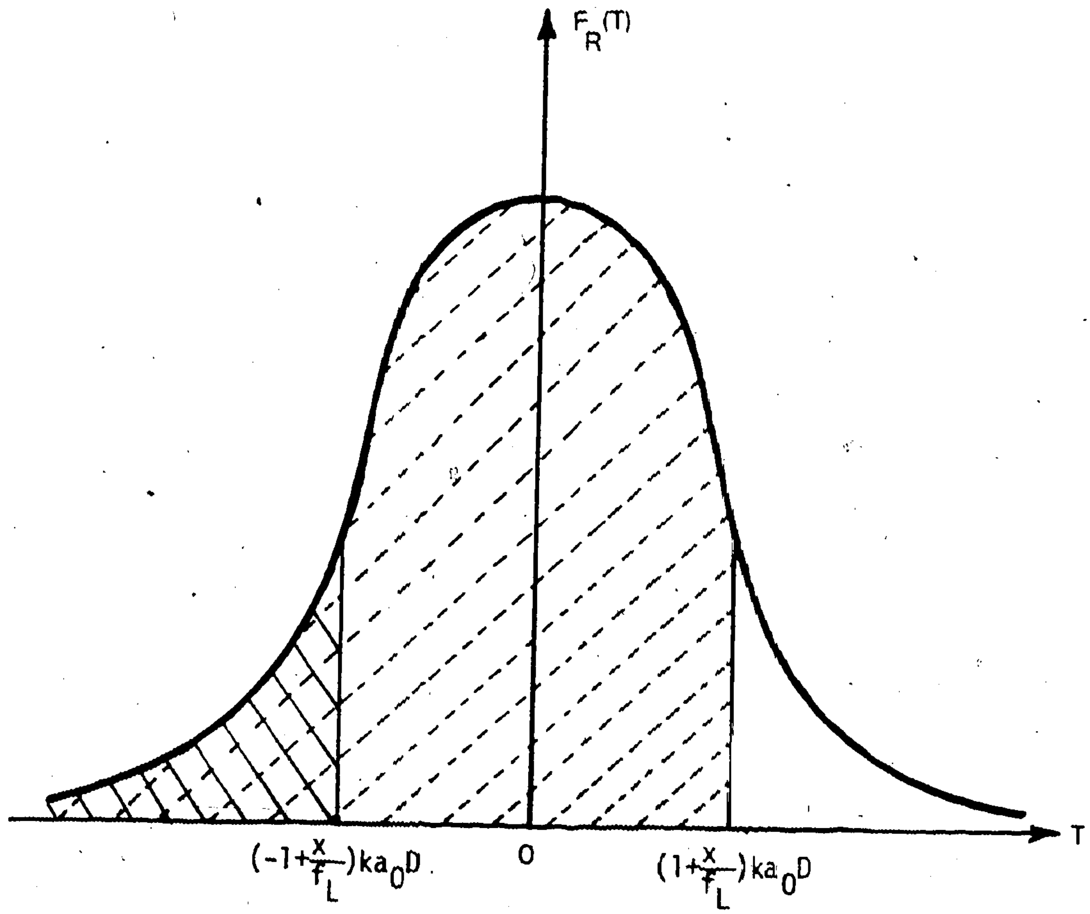
From eq. (5.1.5), the range of values for u_x, u_y in the integrands is between $(-1, 1)$. A comparison of this function with the integrands over u_x, u_y in eq. (5.1.5), shows that as long as the value of t is within the interval $(\pm 1 + \frac{x}{f_L}) k a_0 D$ (or $P_R(t)$ assumes values between A and B), eqs. (5.1.6), (5.1.7) are valid.

Under this scheme, much flexibility is provided for the choice of rays. Since only a finite number of rays are used, a variable ray distribution will prevent a localization of heated regions within the plasma.

5.2 Solenoid grid package

In this section, the choice of grid structure used for defining ray positions is discussed.

In the plasma column, cylindrical symmetry is assumed. A two dimensional numerical grid in radial and axial direction is used. Since this package is designed for implementation in the shell MHD code, the chosen grid structure is the same as that used in that code. In the MHD code, the radial grid layers are made up of coaxial magnetic shells(Fig. 5.2).In this package, the radii of the shells are taken to be the same as the radii



$$F_R(T) = \frac{1}{\sqrt{2\pi}} e^{\frac{-T^2}{2}}$$

$$P_R(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{\frac{-T^2}{2}} dT$$

Area under the curve is A

Area under the curve is B

Figure 5.1 Cumulative gaussian distribution function.

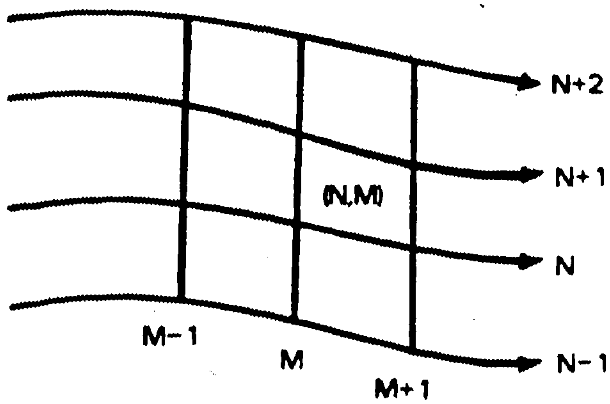


Figure 5.2 Spatial grid structure.

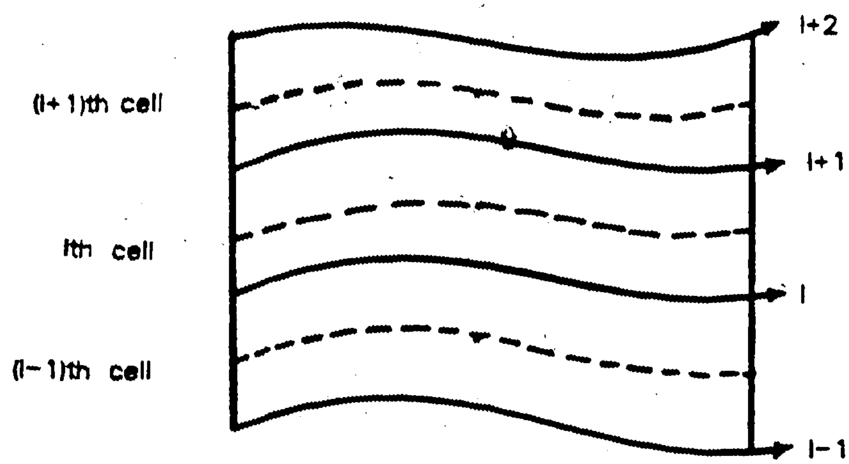


Figure 5.3 Density gradient between shells.

of the magnetic flux shells for the purpose of matching the plasma parameters in the MHD code. Each shell is divided into two subshells so that density gradients can be evaluated in each half of the shell. In the I^{TH} shell, the density gradient in the outer half shell is different from that in the inner half shell due to different values for calculating the gradient. The density gradient in the outer half shell is determined by the difference in densities in the I^{TH} and $(I+1)^{TH}$ shell layer, while the density gradient within the inner half shell is obtained from the difference between densities in the I^{TH} and $(I-1)^{TH}$ shell (fig 5.3).

5.3 Density gradient package

In this section, the fitting of density profiles between any two adjacent grid cells is discussed. By calculating the density gradient changes between grid cells which are radially adjacent to each other, the appropriate profiles can be chosen correspondingly.

From the MHD code, values of plasma density are obtained for the center of each grid cell(see footnote). The radial density gradient between two adjacent cells located at radial distances r_1, r_2 from the axis is determined from the plasma densities N_1, N_2 in these two cells. Once the density gradient of the plasma in a certain region is known, the appropriate density profile can be chosen correspondingly, the solution of the ray equation for this chosen profile can be used to describe the propagation of a ray. In order to determine the proper density profile, the first and second derivative of the density with respect to radial distance, $\frac{dN(r)}{dr}$, $\frac{d^2N(r)}{dr^2}$ are calculated since the choice of profiles is based on the conditions $\frac{dN(r)}{dr}$ >0 or <0 and $\frac{d^2N(r)}{dr^2}$ >0 or <0. The corresponding profiles are listed as follows:(see fig.3.1)

(1) $$\frac{dN}{dr} > 0, \quad \frac{d^2N}{dr^2} > 0, \quad N(r) = N_0\left(1 + \frac{r^2}{a_0^2}\right)$$

(2) $$\frac{dN}{dr} > 0, \quad \frac{d^2N}{dr^2} < 0, \quad N(r) = N_1\left(1 - \frac{a_1^2}{r^2}\right)$$

A cell is bounded by two radial and two axial magnetic shell boundaries.

(3) $\frac{dN}{dr} < 0,\ \frac{d^2N}{dr^2} > 0;\quad N(r) = N_2(1 - \frac{r^2}{a_2^2})$

(4) $\frac{dN}{dr} < 0,\ \frac{d^2N}{dr^2} < 0,\quad N(r) = N_3(1 + \frac{a_3^2}{r^2})$

The parameters N_i, a_i , i=0,1,2,3,can be evaluated by knowing the densities in any two radially adjacent cells(with known radii).

The density curve fitting method, however, does not give a good approximation for the density values at regions where there is a transition from one density profile to another. As shown in figure 5.4, the kind of curve used to approximate the density variation between r_1,r_2 can be $N_0(1+r^2/a_0^2)$ or $N_1(1-a_1^2/r^2)$;similarly in regions between radii r_3 and r_4,profiles $N_2(1-r^2/a_2^2)$ or $N_3(1+a_3^2/r^2)$ can be used. For these regions, profiles used in the shell lying next to this region (in the first case, shell with radius less than r_1 , in the second case, shell with radius less than r_2) are chosen as the fitting profiles.

5.4 Ray tracing package

In this section, methods of tracing the rays through the plasma are discussed. Ray locations at the grid boundaries are computed. Rays staying in the innermost shell are specifically discussed.

This package gives a routine of tracing rays through the plasma column by locating the points which the rays intersect with the cell boundaries. Moreover, absorption coefficient are calculated within each cell.

As soon as the rays reach the column, their radial locations are tracked down to the corresponding grid cell. Knowing which cell the ray hits, its subsequent path can be traced through the application of the ray equation as discussed in section 3.1.

Within each cell, the density is assumed to vary according to one of the profiles given in section 3.1. The subsequent ray path will be governed by the solution corresponding to that particular profile. Ray locations are evaluated at the points where rays intersect with the cell boundaries. At each such intersecting point, the velocity components are evaluated so as to give the initial conditions for the ray entering into another region with a different refractive index.

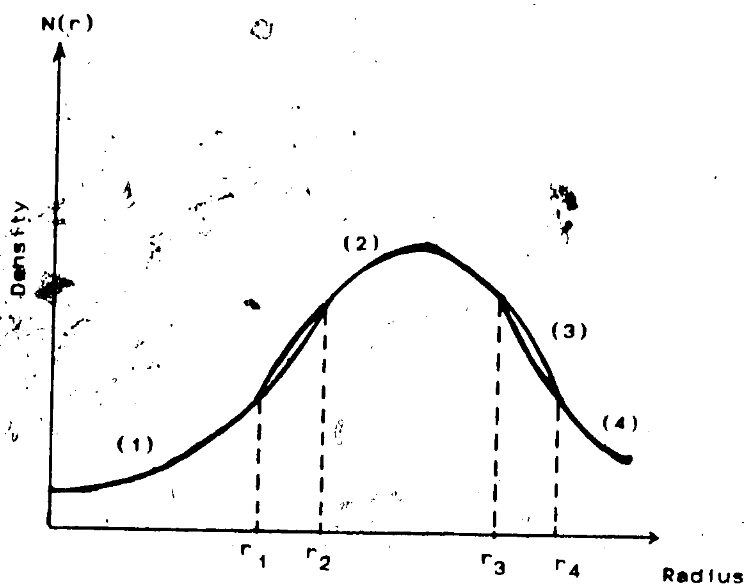


Figure 5.4 Density profile approximation used in various region.

As the initial and final locations of a ray within a cell are known, the associated absorption coefficient can be found from eq. (4.2.2),

$$\int_0^{\Delta s} dK = \int_0^{\Delta s} K_a ds = \int_0^{\Delta t} K_a n(r(t))c dt \tag{5.4.1}$$

where K_a is the absorption coefficient per unit length; Δt is the time for the ray to traverse from the initial position to the final position(or a distance of Δs).

As can be seen from the solutions of the ray equation for different profiles, the radial positions of a ray is a direct function of time. Knowing the time of traverse over a cell, the ray location can be determined. In order to find the traversing time, the final radial position of a ray is taken to be the cell radius(fig. 5.5). Corresponding time is calculated from the equations derived in section 3.1. The ray advances an axial distance of $v_z\Delta t$ in a time interval of Δt. If this axial distance is less than an axial grid interval, it indicates that the ray will cross the radial grid boundary before it crosses the axial grid boundary (fig. 5.4). On the other hand, if the calculated axial distance is larger than the axial grid separation, the ray will cross the axial grid boundary before it crosses the radial one. The time is then changed to that for the ray to travel over one axial grid distance. The radial distance corresponding to this time can be evaluated from the ray solution. Axial and radial distances are thus known. The angular position can also be found from the equation

$$r^2\dot{\theta} = r_0^2\dot{\theta}_0$$

$$\theta = \theta_0 + \int_0^{\Delta t} \frac{r_0^2\dot{\theta}_0}{r^2} dt \tag{5.4.2}$$

where $r_0, \dot{\theta}_0$ are the initial radial ray location and angular velocity respectively. The three spatial co-ordinates of a point on a trajectory can be completely defined.

When a ray reaches the inner most shell, the ray path remains straight since there is no change in refractive index within the core and the ray is not refracted. Ray locations at the boundary points and at the closest point to the axis have to be found differently. The ray path is illustrated in fig. 5.6.

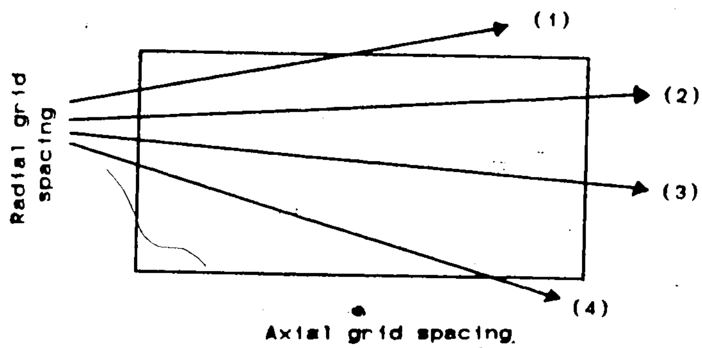


Figure 5.5 Intersection points of rays with cell boundaries.

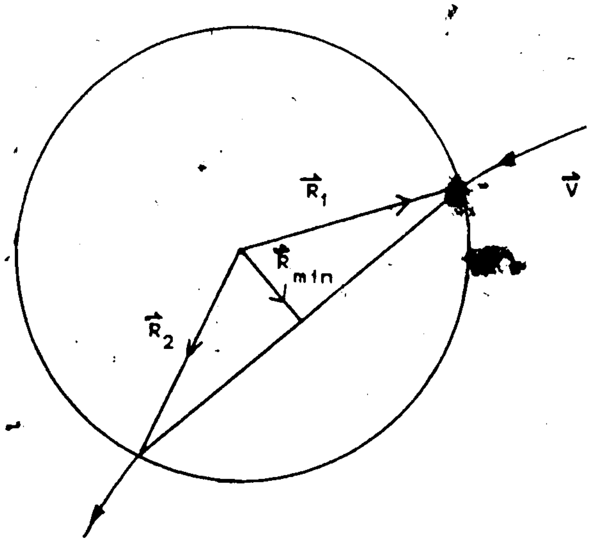


Figure 5.6 Ray path in the central core of the plasma column.

It should also be noticed that the radial velocity component is taken to be negative if the ray travels towards the axis, otherwise positive. The absorption path for calculating the absorption coefficient is the magnitude of the ray segment between the initial and final ray locations within a cell. The time taken for a ray to traverse from one side of the core boundary to another side is

$$t = \frac{-2\vec{R}_1 \cdot \vec{v}}{|\vec{v}|^2} \tag{5.4.3}$$

where $\vec{R}_1$ is the radial position of the ray at the shell boundary; $\vec{v}$ is the velocity within the core and at the boundary. The time taken for a ray to traverse from the boundary to a point which is closest to the axis is

$$t_{min.dist.} = \frac{-\vec{R}_1 \cdot \vec{v}}{|v|^2}. \tag{5.4.4}$$

With this time and the velocity at the boundary of the core, locations of the closest point can be determined as

$$\vec{R}_{min} = \vec{v}\, t_{min.dist.} + \vec{R}_1 \tag{5.4.5}$$

Depending on the velocity components of a ray, there are three types of rays. Rays with a non-zero angular velocity will spiral in a trajectory forming a helix around the axis of propagation and approach a position close to the axis but never through it because of the finite angular momentum of the ray. Rays without an angular velocity component, will go through the axis. Finally, rays that have only a non zero axial velocity component will propagate along the axis.

5.5 Absorption package

This package calculates the power absorption along a ray. Power exchanges are only encountered in those cells through which rays pass.

Power absorption in the I^{TH} cell can be calculated from eq. (4.2.7), namely,

$$\Delta P_{abs} = P_i(r_0, t_0)\left[1 - e^{-\int_0^{\Delta t_i} K_a n(r(t)) c\,dt} \right] \tag{5.5.1}$$

where $P_i(r_0, t_0)$ denotes the power carried by the ray before it traverses the I^{TH} cell; t_0 is the time at which the ray advances to the I^{TH} cell; $r_0(t_0)$ is the radial position of the ray before it enters the I^{TH} cell; Δt_i is the time taken to traverse through the cell.

The temporal power profile of the laser beam follows the one used in the MHD code. It is approximated as straight line segments as shown in fig. 5.7. In the MHD code, the temporal change of the plasma behaviour is calculated at discrete time values(or hydrodynamic time steps, Δt_{MHD})(see footnote). Power is assumed to remain constant within a hydrodynamic time step. It is only changed when the hydrodynamic time steps are altered. This assumption is valid as long as the size of a time step is much less than the pulse width. For the case of a short solenoid (5cm in length,ion and electron temperature of 1eV), the step size is in the order of 10^{-8} sec. and the pulse width is in the order of 10^{-6} sec. Thus, the assumption is a reasonable approximation. As the plasma becomes hotter, the step size even gets smaller since the particles become more energetic and their velocities thus become higher.

The advancement of the beam is simulated in terms of a number of beam time steps, Δt_{LASER}, which is the time for the beam to propagate over one cell. Within one hydrodynamic time step, the number of beam time steps is

$$M = \frac{\Delta t_{MHD}}{\Delta t_{LASER}} \tag{5.5.2}$$

The beam advances over one cell spacing for every beam time step.

The hydrodynamic time step(Δt_{MHD}) is the time interval within which the plasma is taken to be stable. This time is bounded by the time for the plasma fluid element to traverse over a grid cell radially or axially. Namely, $\Delta t_{MHD} = \Delta R / v_{ma}$ or $\Delta t_{MHD} = \Delta x / v_s$ where $\Delta R, \Delta x$ are the radial and axial grid size; v_{ma} is the magneto-acoustic velocity, $\sqrt{(v_s^2+v_A^2)/(1+v_A^2/c^2)}$; v_A is the Alfven velocity, $B^2/\sqrt{4\pi N_{ion}\, m_{ion}}$; v_s is the sound velocity, $\sqrt{T_e/m_e}$. In actual runs, the time step is adjusted to be less than 50% of these limits.

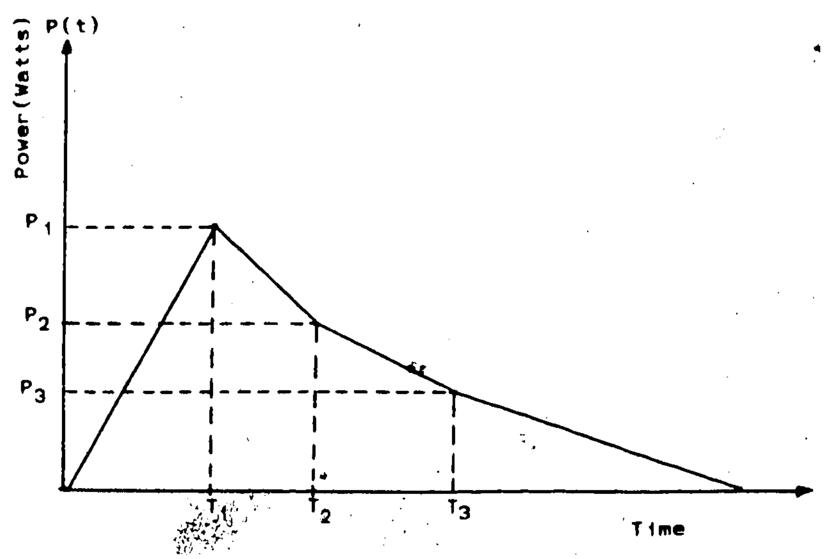


Figure 5.7 Temporal power profile of the laser beam.

The absorbed energy in a cell can be calculated as follows. The initial energy carried by a ray in any hydrodynamic time step is

$$E_{MHD} = \Delta t_{MHD} \times P_{MHD} \tag{5.5.3}$$

where P_{MHD} is the laser power of the ray in a particular MHD step. Having this energy divided among the M beam time steps, the initial energy which the ray carries is

$$E_{LASER} = \frac{\Delta t_{MHD} \times P_{MHD}}{M}$$

$$= \Delta t_{LASER} \times P_{MHD} \tag{5.5.4}$$

This amount of energy will be the input energy for each laser time step. The energy deposited in an individual cell will be governed by an equation similar to eq. (5.5.1), that is,

$$E_{abs} = E_i [1 - e^{-\int_0^{\Delta t} K_a n(r(t)) c dt}] \tag{5.5.5}$$

where E_i is the energy associated with the ray before it enters into the I^{TH} cell and is equal to $E_{i-1}\, e^{-\int_0^{\Delta t} K_a n(r(t)) c dt}$. The total amount of energy absorbed in a cell will be the sum of all the absorbed energy contributed from each ray passing through the cell.

5.6 Ponderomotive force package

In this section, the ponderomotive force in each cell through which the ray passes is computed from the energy intensity gradient across the grid cells.

From eq. (4.3.11), (4.3.12), the radial and axial force components are given as

$$(F_{NL})_r = \frac{-\omega_{pe}^2}{2\omega_0^2} \frac{\partial}{\partial r}\left(\frac{I}{nc}\right) \tag{4.3.11}$$

$$(F_{NL})_z = \frac{-\omega_{pe}^2}{2\omega_0^2} \frac{\partial}{\partial z}\left(\frac{I}{nc}\right) \quad (4.3.12)$$

Through using finite difference scheme, the force components in the cell labelled with grid co-ordinates (N,M) are

$$(F_{NL})_r = \frac{-\omega_{pe}^2}{2\omega_0^2} \frac{I_{Upper} - I_{Low}}{R_{N+1} - R_N} \frac{1}{\eta_{N,M}c} \quad (5.6.1)$$

$$(F_{NL})_z = \frac{-\omega_{pe}^2}{2\omega_0^2} \frac{I_{Right} - I_{Left}}{Z_{M+1} - Z_M} \frac{1}{\eta_{N,M}c} \quad (5.6.2)$$

where I_{UPPER}, I_{LOW}, I_{RIGHT}, I_{LEFT} are radiation intensities at the cell boundaries located at radial distances R_{N+1}, R_N, and axial distances Z_{M+1}, Z_M, respectively; $\eta_{N,M}$ is the refractive index in cell located at grid position (N,M) (fig. 5.1).

The intensities at the cell boundaries are taken to be the averages of the cell centre values of two adjacent cells. Thus, for cases of I_{UPPER} and I_{LOW}, they are given as

$$I_{Upper} = \frac{1}{2}\left(I_{N+1,M} + I_{N,M}\right)$$

$$I_{Left} = \frac{1}{2}\left(I_{N,M} + I_{N,M+1}\right) \quad (5.6.3)$$

where N,M are subscripts referred to the corresponding grid cell.

For cells located at the plasma boundary, they are not completely surrounded by other cells. At the outermost radial boundaries, the intensities are taken as half the value at the centre. At both ends of the column, the intensities at the outermost boundary are taken to be that of the beam.

Chapter 6

Computational results

In section 6.1, plots for ray trajectories in a vacuum, and plots for the spatial distribution of ray locations and energy are given and discussed. In section 6.2, the simulation of a typical plasma density profile from the data obtained from the shell MHD code is discussed. Ray trajectories are computed within a plasma column for the assumed density profile in section 6.3. The behaviour of the rays within the medium is also discussed. In section 6.4, the energy distribution in the plasma is presented in terms of the ray distribution at various axial positions along the column. Finally, the absorbed energy and ponderomotive forces within the plasma are illustrated with the three dimensional plots.

6.1 Ray propagation in vacuum

In this section, trajectories of rays propagating in vacuum , the spatial distributions of the rays and the radial variation of beam power are plotted and discussed.

A sample of 100 rays is used to simulate the beam. Beam power is divided equally among the rays. A lens of 5cm(radius) aperture and a focal length of 150cm is chosen. Transverse locations and directions of the rays at the lens plane are determined from a random Gaussian distribution function as explained in section 5.1. The locations of the rays along the direction of propagation are determined from eq. (2.1.26) given in chapter two. The ray trajectories in vacuum are shown graphically in fig. 6.1 from which it is clear that the density of rays is higher in the axial region. Such a ray distribution results from a Gaussian choice of the intensity profile for the beam. The nonzero radial position of the rays at the focus(shown in Fig. 6.1) is due to the inclusion of diffraction effects.

A plot of the square of the average radial distance of all rays is given as a function of axial distance in fig. 6.2. The trajectory shows axial symmetry about the focus. In fig. 6.3, there is a minimum for the square of the half power beam radius at the focal distance. The half power beam radius at the focus is calculated to be 4.88 x 10^{-3}cm. This finite beam size is a result of the inclusion of diffraction effects.

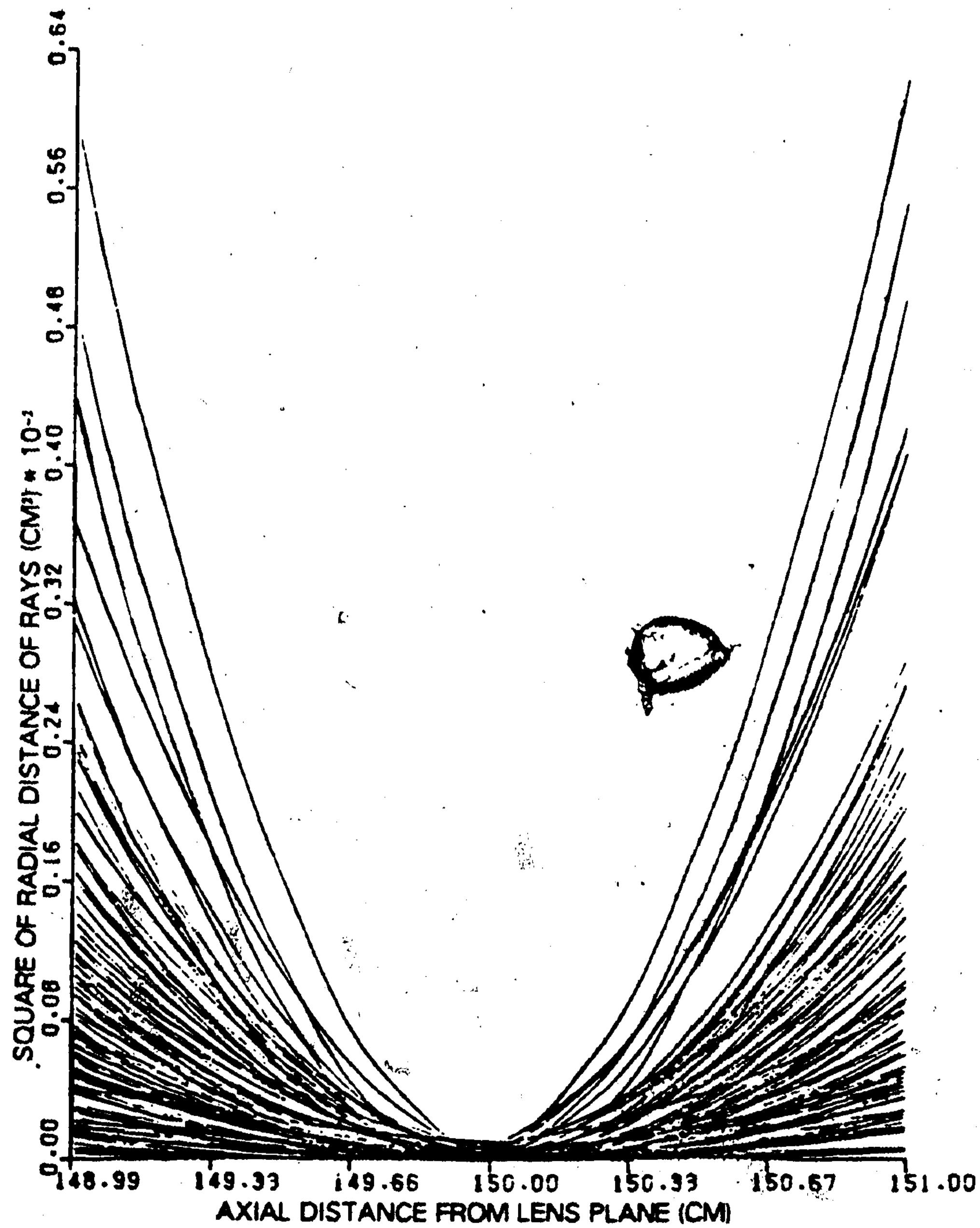


Figure 6.1 Variation of the square of radial position of the trajectories along the axis of propagation.

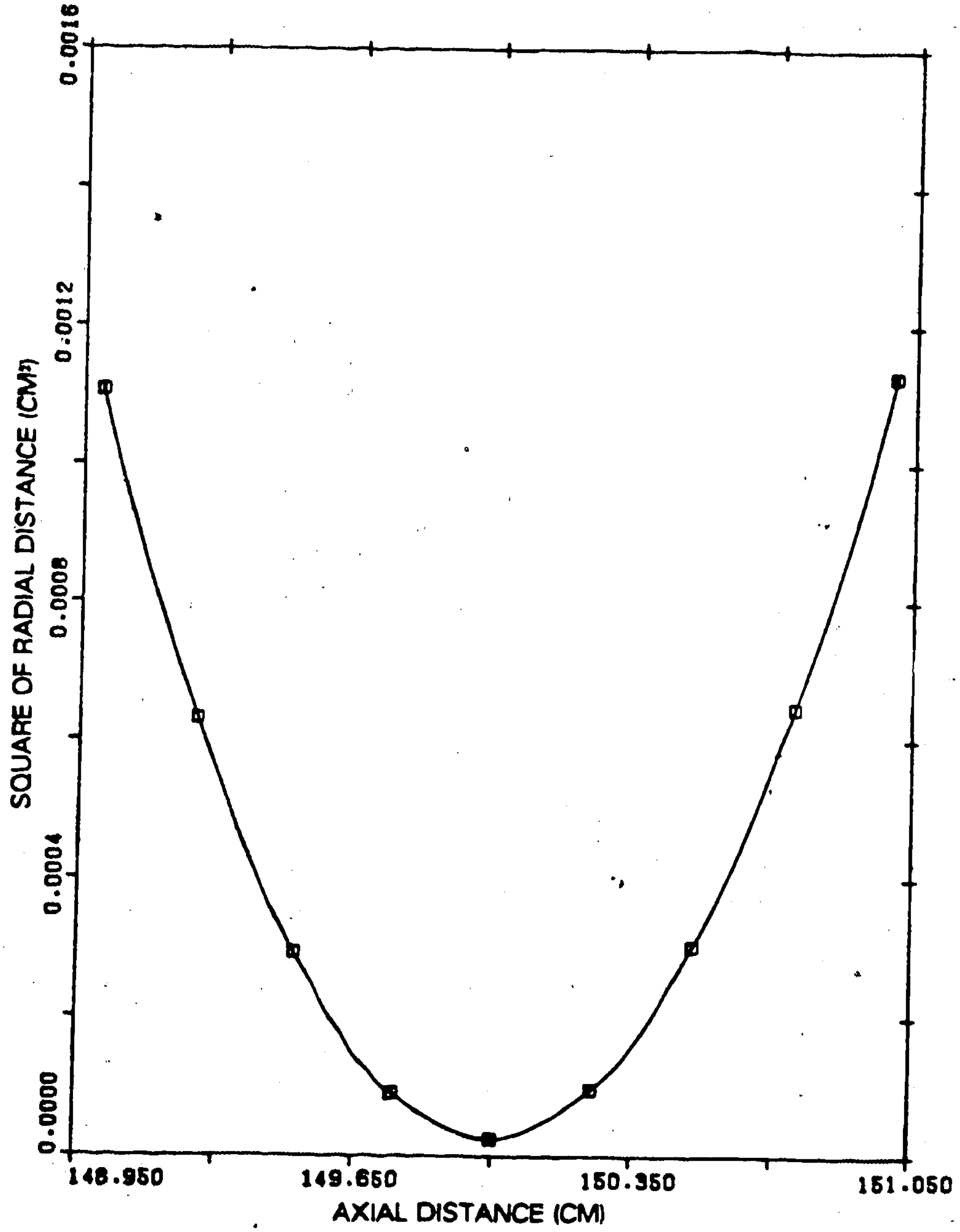


Figure 6.2 Square of the average radial ray position with incoherence factor=0.5.

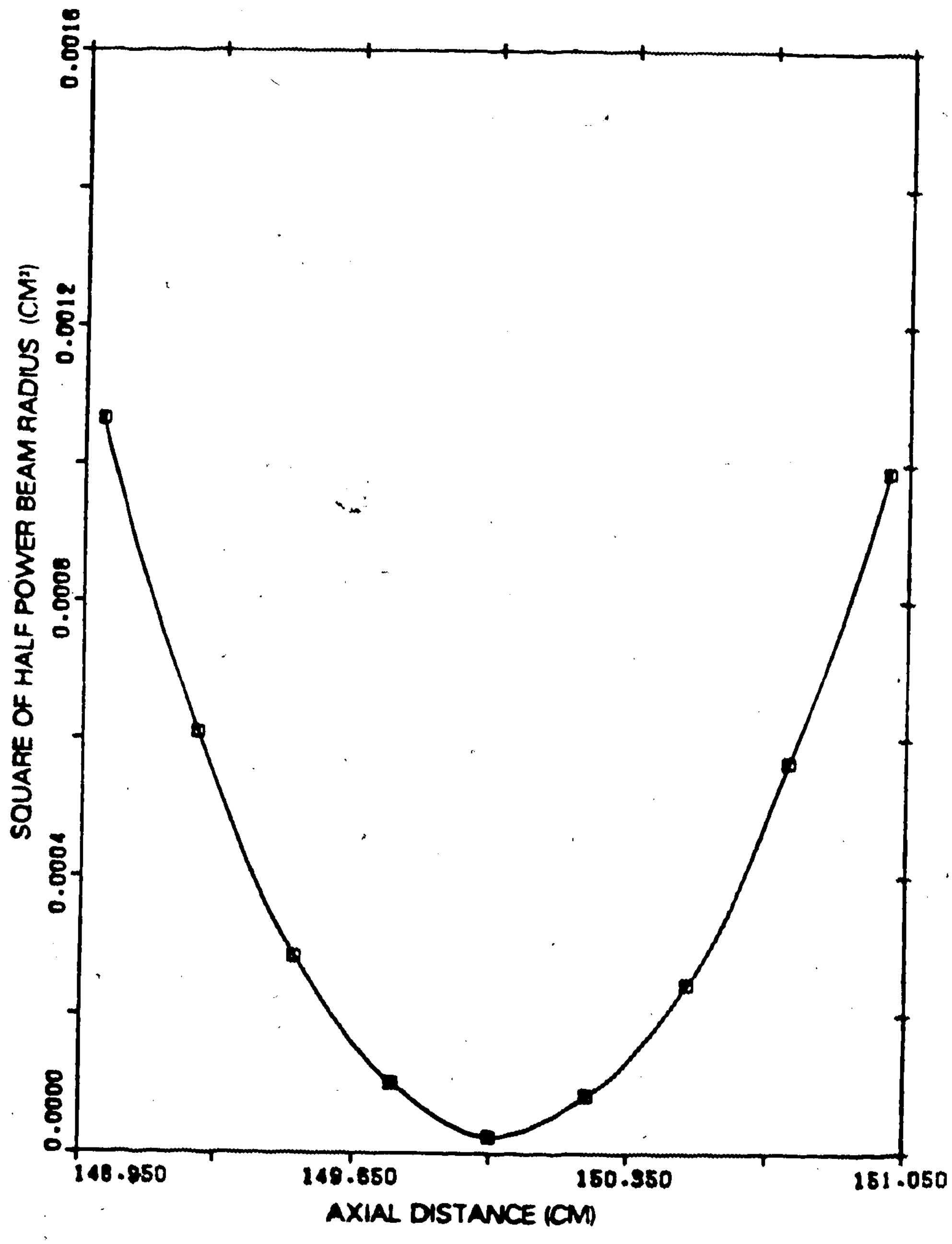


Figure 6.3 Square of the half power radial ray position around focal spot with incoherence factor= 1.0.

The distribution of ray locations at the lens plane and the focal plane are shown in figs. 6.4 and 6.5 respectively. According to the diagrams, the spread of rays at the focal plane is shown to have a similar pattern as that at the lens plane. This implies that the locations of rays have a Gaussian distribution at the focal plane. An analysis on the radial power distribution confirms the above conclusion.

The power distribution profile at the lens plane and the focal plane are obtained by counting the number of rays within a circular area of radius r. Since each ray carries equal units of power, the total number of rays represent the total units of power within the region. This is presented as cumulative histograms in figs. 6.6 and 6.7. The average deviation of the power values from those calculated directly from a Gaussian intensity profile is about 0.3%, indicating that the rays at the focal plane follow a Gaussian distribution.

The square of the average radial distance of the rays around the focus for an incoherent beam (incoherence factor=0.5) is shown in fig. 6.8 and the half power radius is given in fig. 6.9. The spotsize (radius 9.77×10^{-3}cm.) is about twice as much as that for the coherent beam case. A plot of the power distribution is also given in fig. 6.10. The degree of incoherence can be used as a means for altering the size of the beam at the focus.

6.2 Simulation of the density profile

In this section, a plasma density profile used for testing the ray tracing routine is constructed from the density values computed from McMullin et. al.'s MHD code for a short laser heated solenoid(5cm. in length and .5cm. in radius). The plasma is assumed to have an initial density of 2.0×10^{19} cm^{-3} and is confined by a magnetic field of 100 kilogauss. The laser power is assumed to rise linearly from 0 at t=0 to 100MW at t=10ns and then remains constant. A typical density profile is constructed from the density values computed at time 1.2×10^{-7}sec. The density values at an axial distance of 0.917cm. from the laser entry end are listed in the following table:

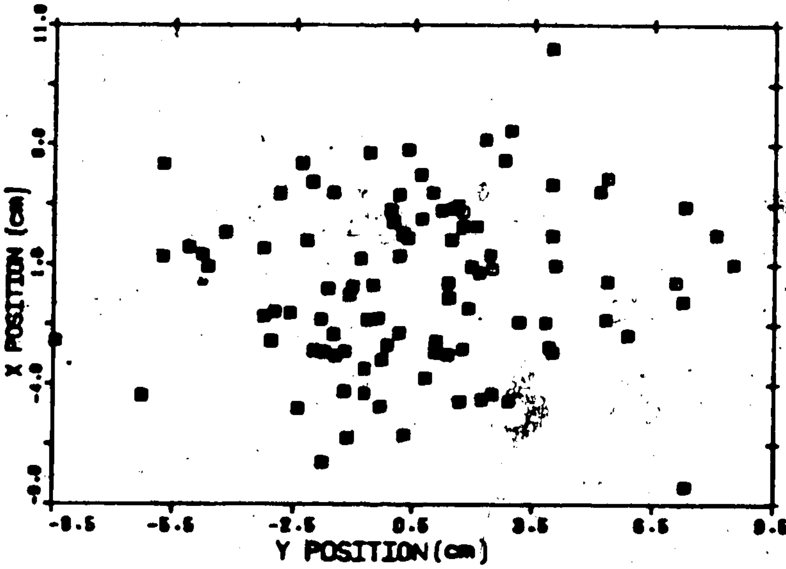


Figure 6.4 Distribution of rays at the lens plane (100 rays)

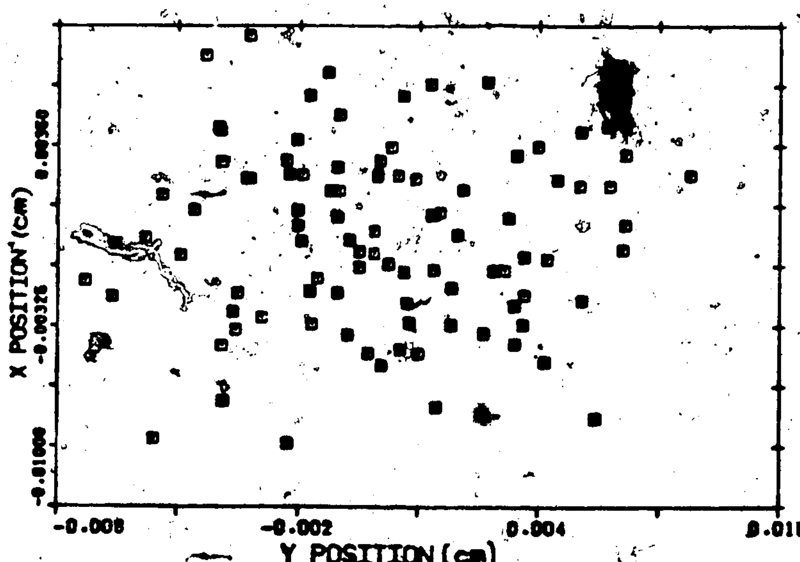


Figure 6.5 Distribution of rays at the focal plane (100 rays).

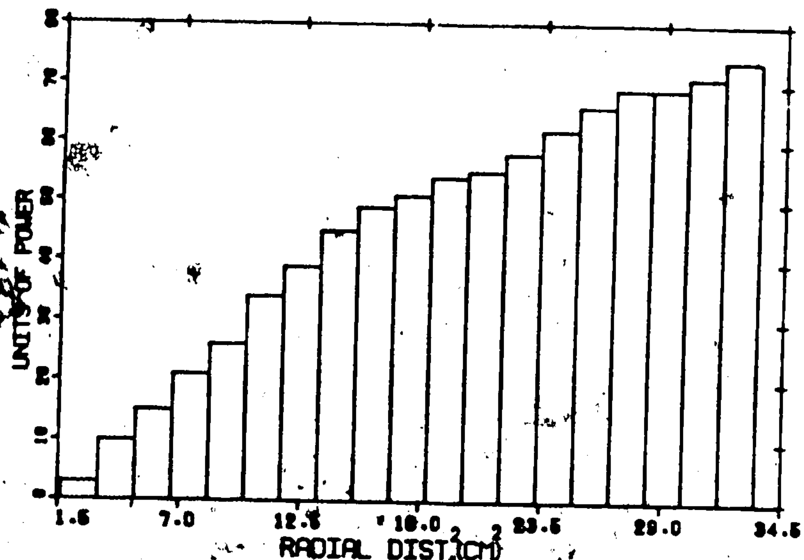


Figure 6.6 Power distribution for beam with incoherence factor=1.0 at the lens plane

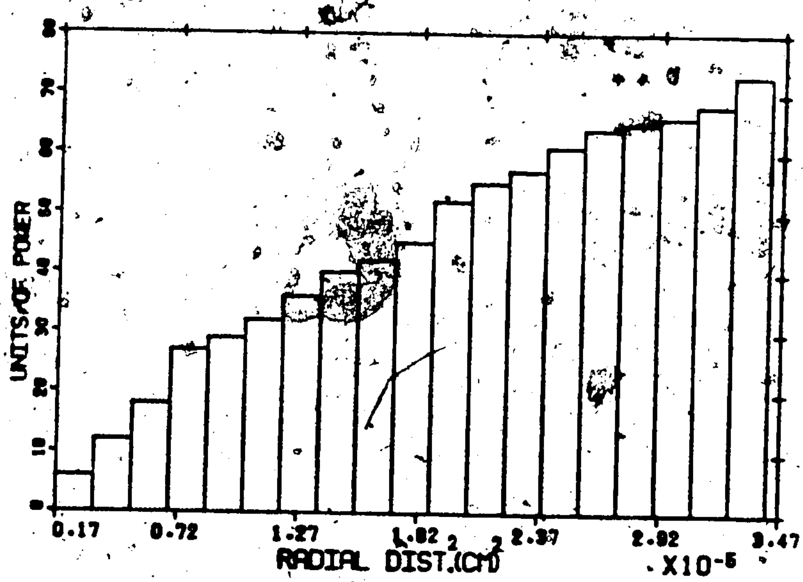


Figure 6.7 Power distribution for beam with incoherence factor=1.0 at the focal plane.

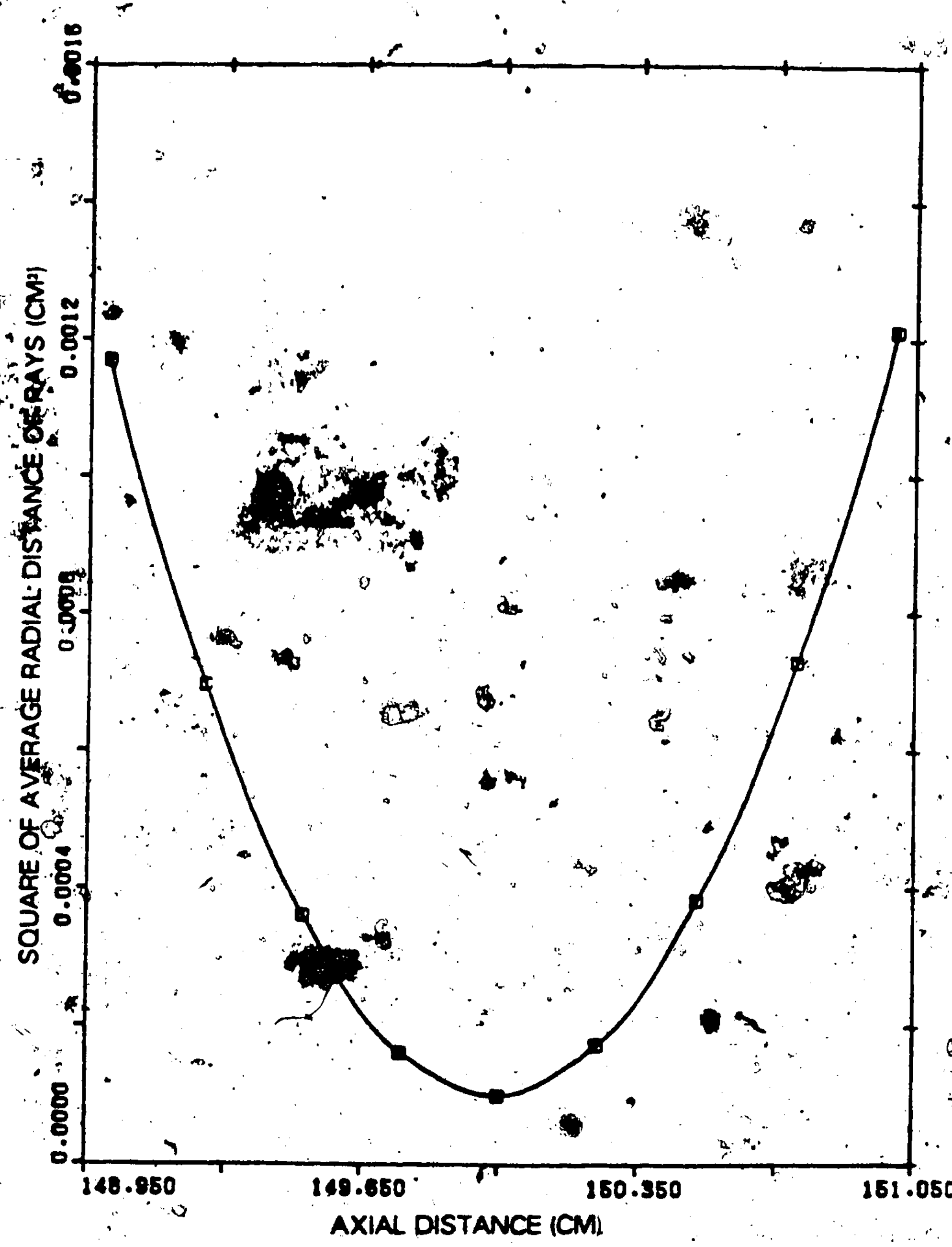


Figure 6.8 Square of average radial distance of rays around the focus.

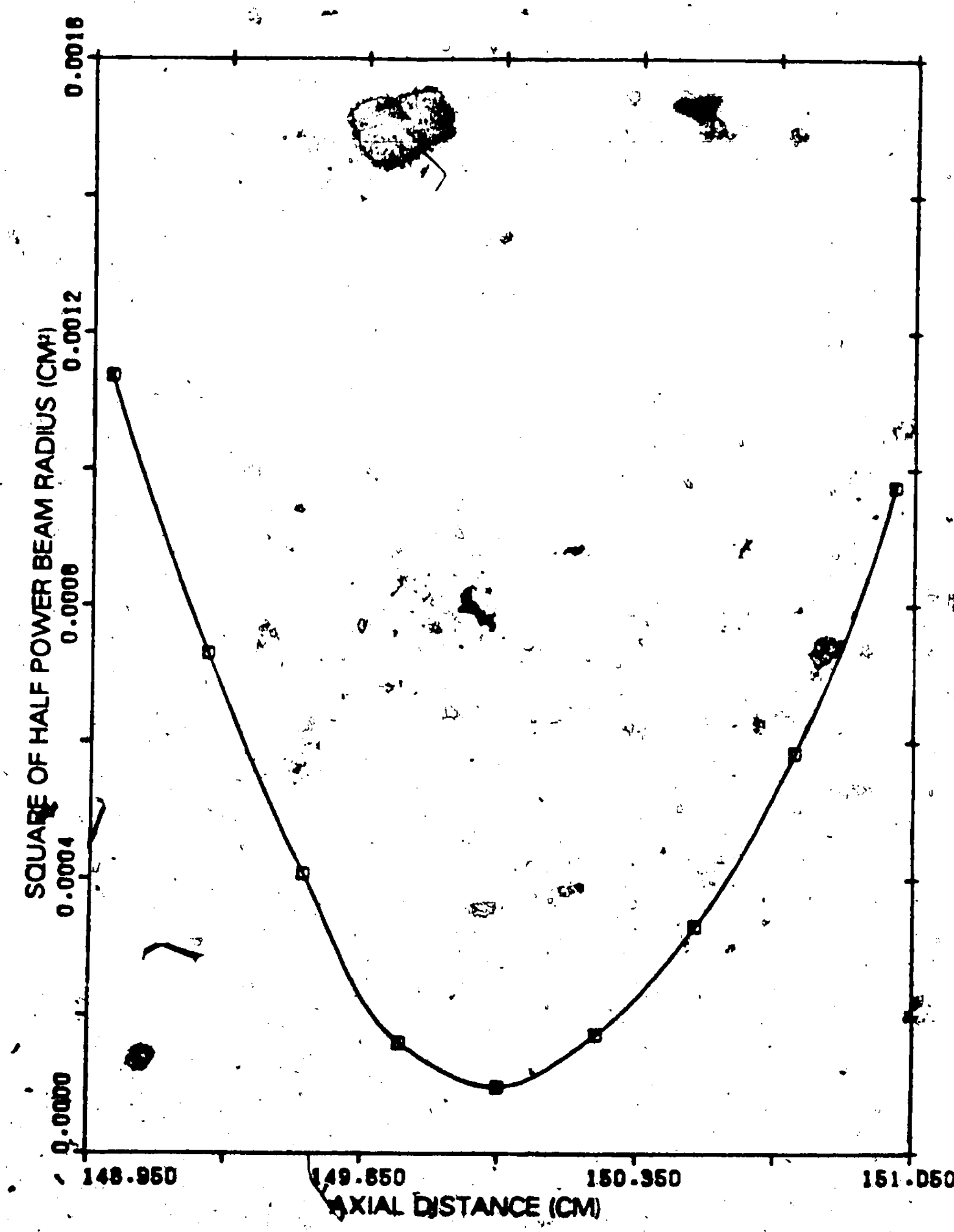


Figure 6.9 Square of half power beam radius with the incoherence factor=0.5.

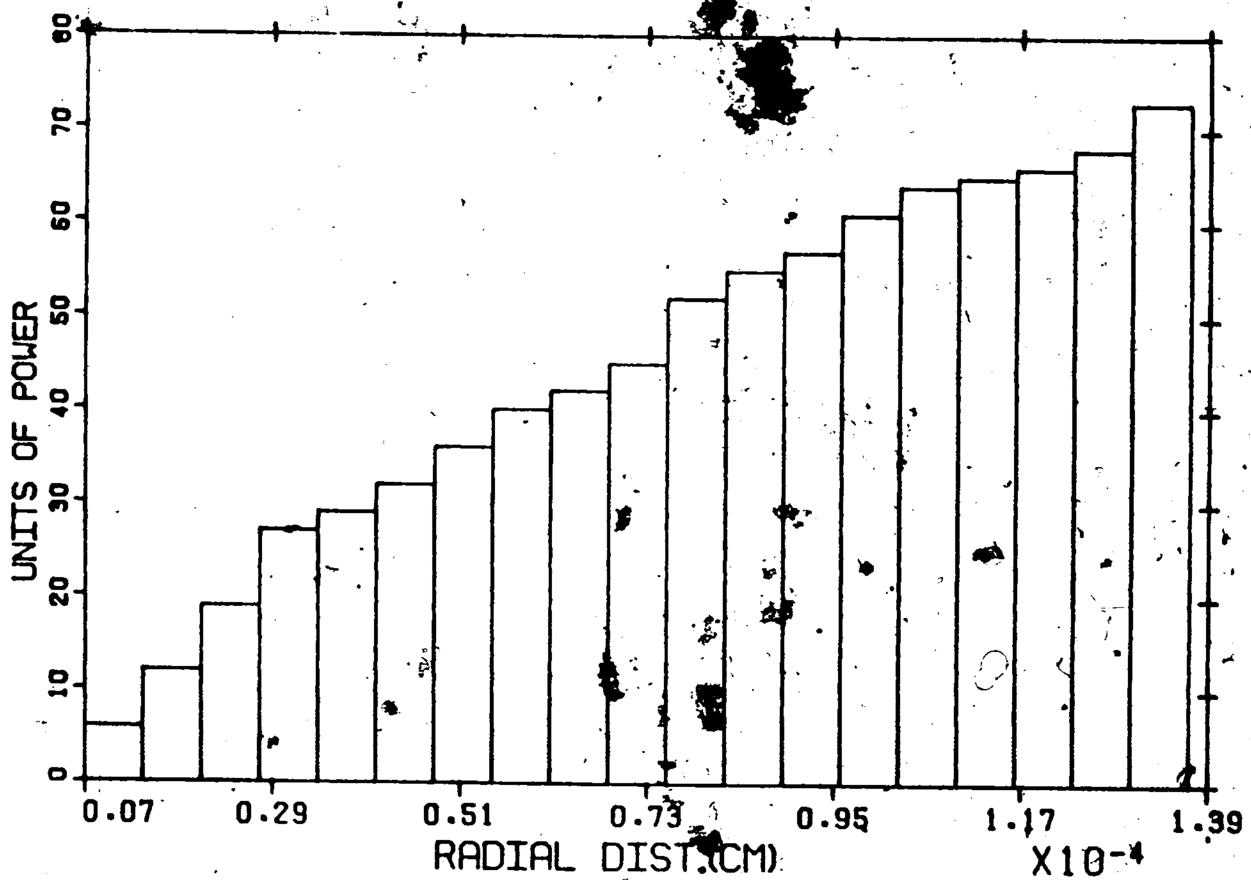


Figure 6.10 Power distribution for beam with the incoherence factor=0.5.

Table 1

	Radial position(cm)	Density(x $10^{11}/cm^3$)
(1)	0.1330	1.152
(2)	0.2745	1.656
(3)	0.42[illegible]	2.46
(4)	0.6720	2.26
(5)	0.9740	2.0

The density variation between the first and second locations is fitted with the profile $N_0(1+r^2/a_0^2)$. For the second and third locations, the profile $N_1(1-a_1^2/r^2)$ is used. Following the same procedure, the rest of the data are fitted with the corresponding profiles $N_2(1-r^2/a_2^2)$ and $N_3(1+a_3^2/r^2)$. This simulated profile is illustrated in fig. 6.11.

6.3 Ray Tracing in the plasma column

In this section, the radial locations of rays propagating in various regions of the plasma column [illegible]lotted and discussed. The radial components of the rays are calculated according to the ray eq[illegible]solutions for the density profiles in various regions.

1. *Rays with radially outward velocity components*

The radial components of the ray trajectories within the plasma for rays having an initial radial outward velocity are shown in figs. 6.12 to 6.17. The plasma column is assumed to be placed 10cm behind the focus to ensure that the rays will diverge before they enter into the medium.

In fig. 6.12, the radial component of the ray trajectory within a radial parabolic density profile(see region 1 in fig. 6.11) is shown. The initial transverse co-ordinates and directions of the ray are x=0.219cm, y=−0.114cm, u_x=0.0147, $u_y = -0.715 \times 10^{-4}$. The sinusoidal fluctuation of the radial component of the ray along the plasma column shows that the ray is trapped within the region. A full illustration of the ray propagating along the column is given in fig. 6.13. The ray gyrates around the axis of propagation and traces a helix with an oscillating radius.

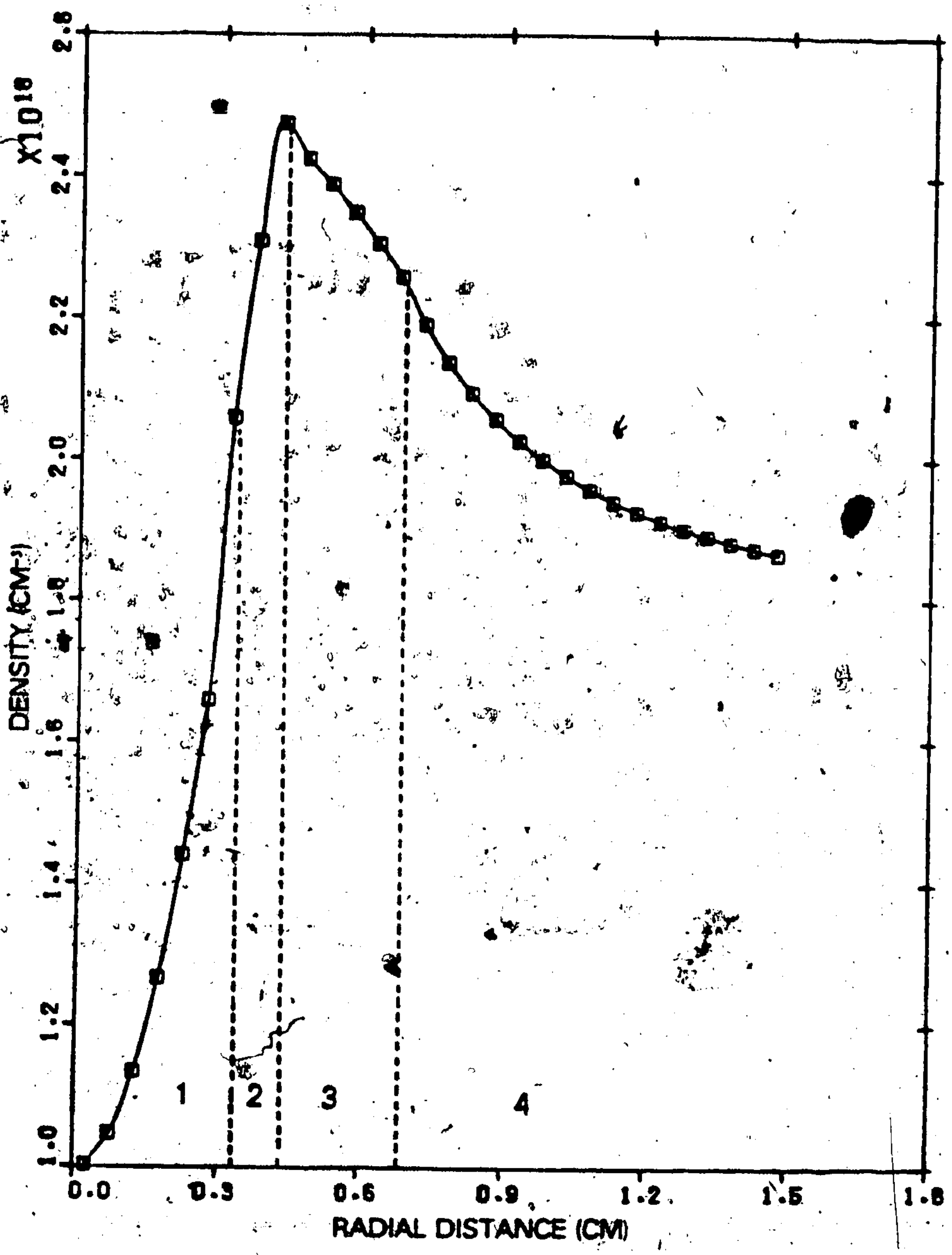


Figure 6.11 Radial density profile of the plasma column.

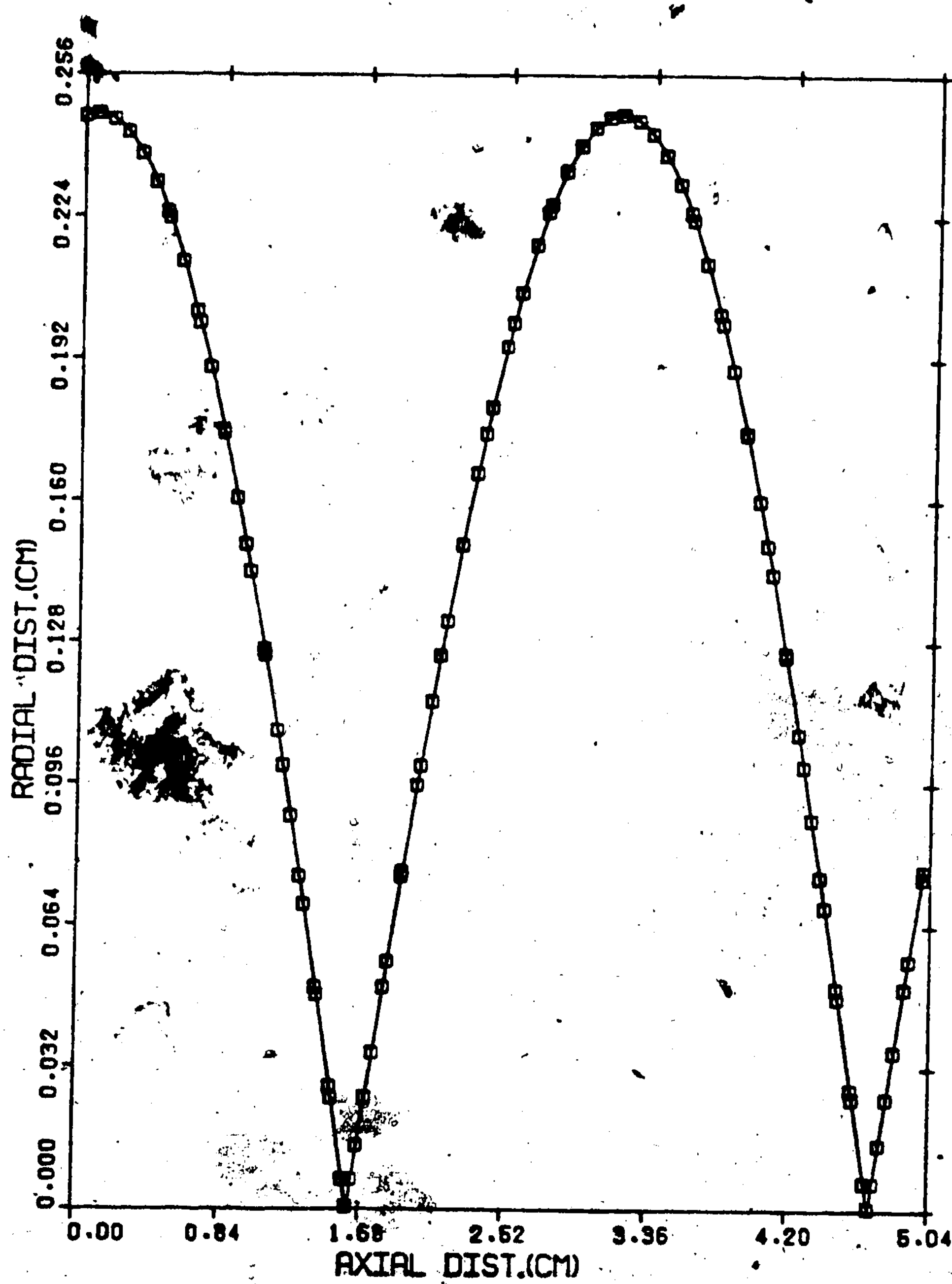


Figure 6.12 Ray path within region 1 (with initial outward radial velocity)

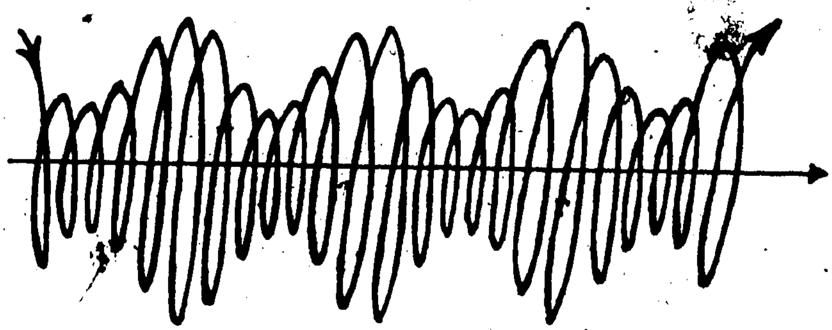


Figure 6.13 Ray trajectory in plasma column

This gyrating ray trajectory shows that beam focusing and defocusing takes place in the parabolic density region.

In the following, the result obtained by the ray tracing technique is compared with those obtained from the normal mode analysis developed by McMullin,Capjack,and James. From the normal mode analysis, the axial period of beam intensities within a parabolic density profile is

$$K = \left.\frac{\omega_{pe}(r,z)}{k(z)c_0 a_0(z)}\right|_{z=0,r=0} \qquad (6.3.1)$$

where

$$\omega_{pe}^2(r,z)\Big|_{\substack{r=0\\ z=0}} = \frac{4\pi e^2 N_e(0,0)}{m_e}$$

$$k(z)_{\substack{r=0\\ z=0}} = \frac{2\pi}{\lambda}\left[1 - \frac{\omega^2}{\omega_{pe}^2(0,0)}\right]^{\frac{1}{2}}$$

c_0 is the speed of light in vacuum.

$a_0(z)$ is the coefficient used in the following density profile,

$$N_e(r,z) = N_0(0,z)\left[1 + \frac{r^2}{a_0^2(z)}\right]$$

and $N_e(0,z)$ is the axial density.

For the special case of axially independent plasma density, eq.(6.3.1) becomes

$$K = \left.\frac{\omega_{pe}(r)}{k(0)c_0 a_0(0)}\right|_{r=0} \qquad (6.3.2)$$

By substituting for $N_0 = 0.997 \times 10^{11}/cm^{-3}$, $a_0(0)=0.338$ and $\omega_{pe}(0)=5.63 \times 10^{13}\ sec^{-1}$ in the above equation, the axial period of oscillation is found to be 6.028cm. From the ray tracing computation,the period is calculated to be 6.1678cm which is within 2% error with the value calculated from eq. (6.3.2). The period obtained from the ray tracing technique is further compared with that derived by Mani. Results show a small discrepancy of 2.5%. Thus, this ray tracing method gives consistent description of the beam propagating in a medium with a parabolic density profile.

In fig. 6.14 and fig. 6.15, the radial variations of two rays propagating in the region where the plasma density varies according to the relation $N_1(1-a_1^2/r^2)$ (see region 2 in fig. 6.11) are shown. The case in which the ray penetrates into a region where the plasma density is close to the peak value is displayed in fig. 6.14. The initial locations and directions of the ray are $x=-0.396$cm, $y=0.149$cm, $u_x=-0.0265$, $u_y=0.0103$. The plasma density in this region is too high for the ray to be trapped. As a result, the ray propagates radially outward and enters into another region with a radially decreasing plasma density, where the ray is further refracted off the column.

In fig. 6.15, the ray propagates close to region 1 (see fig. 6.11). The initial locations and directions are $x=0.091$cm, $y=0.295$cm, $u_x=0.646 \times 10^{-2}$, $u_y=0.199 \times 10^{-1}$. The plasma density is high enough to cause total reflection of the ray. Consequently, the ray penetrates into the parabolic density region where it is trapped.

In fig. 6.16, the radial component of the ray is seen to increase as the ray propagates within region 3 (see fig. 6.11). The initial transverse co-ordinates and directions of the ray are $x=-0.22$cm, [illegible], $u_x=-0.$[illegible]45, $u_y=-0.0265$. The ray is refracted off the plasma column due to a [illegible] in the refractive index.

The variation of the radial component illustrated in fig. 6.17 gives how a ray propagates along the plasma column if it initially lies close to the plasma periphery (see region 4 in fig. 6.11). The ray locations and directions are chosen to be $x=0.822$cm, $y=-0.731$cm, $u_x=0.0551$, $u_y=-0.0487$. The plot shows that the ray propagates only within the outside core of the column and cannot penetrate into the plasma column.

2. *Rays with radially inward velocity components*

In this section, rays with an initial radially inward velocity are traced along the plasma column in various density regions. The plasma column is assumed to be placed 10cm in front of the focus. Rays are thus ensured to be converging by the time they reach the column.

In fig. 6.18, the radial component of the ray location in the parabolic density region (see region 1 in fig. 6.11) is shown. The initial locations and directions are $x=0.287$cm, $y=0.69 \times 10^{-1}$cm, $u_x=-0.192 \times 10^{-1}$, $u_y=-0.465 \times 10^{-3}$. The period of

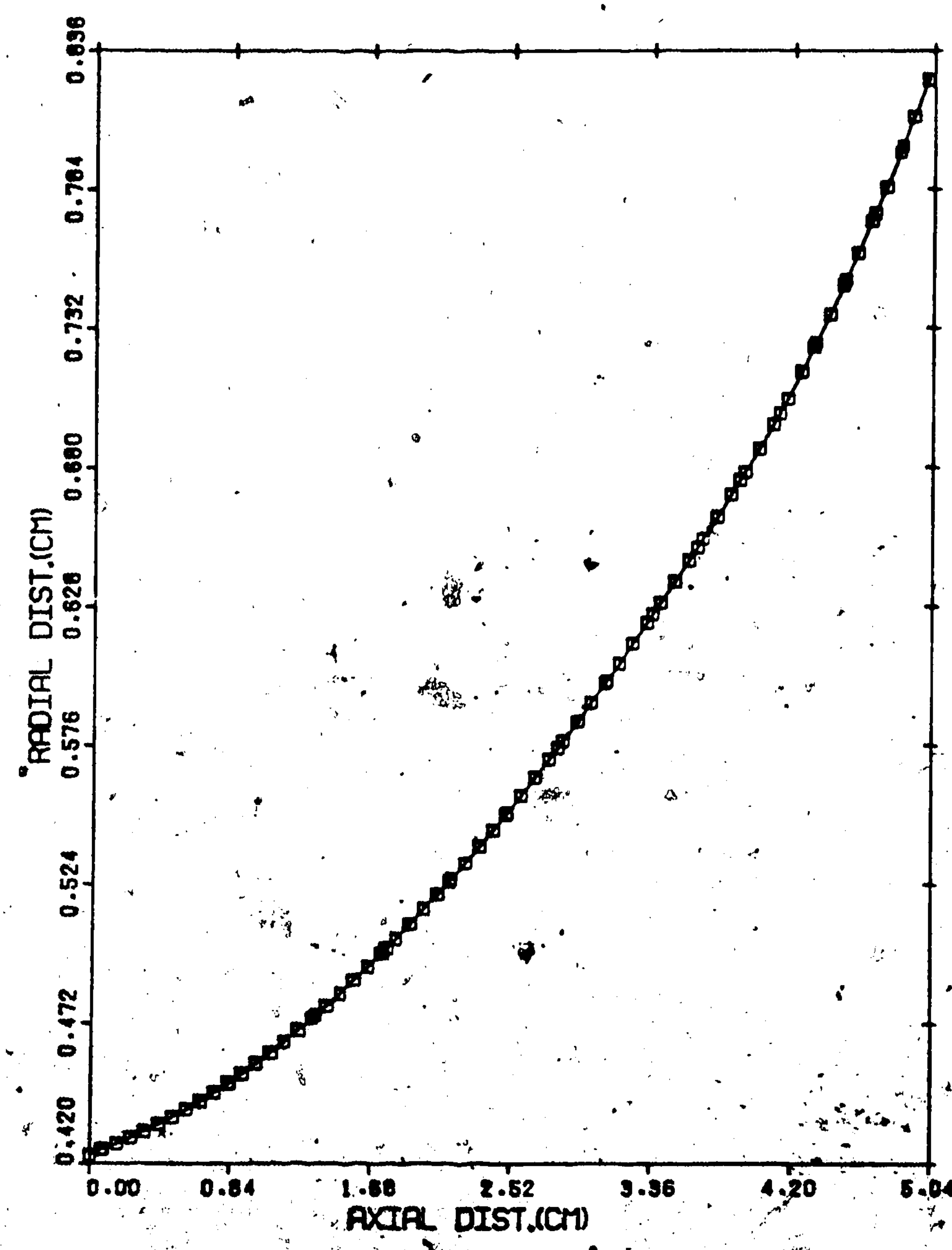


Figure 6.14 Ray path within region 2 (with initial outward radial velocity and initial position close to region 3).

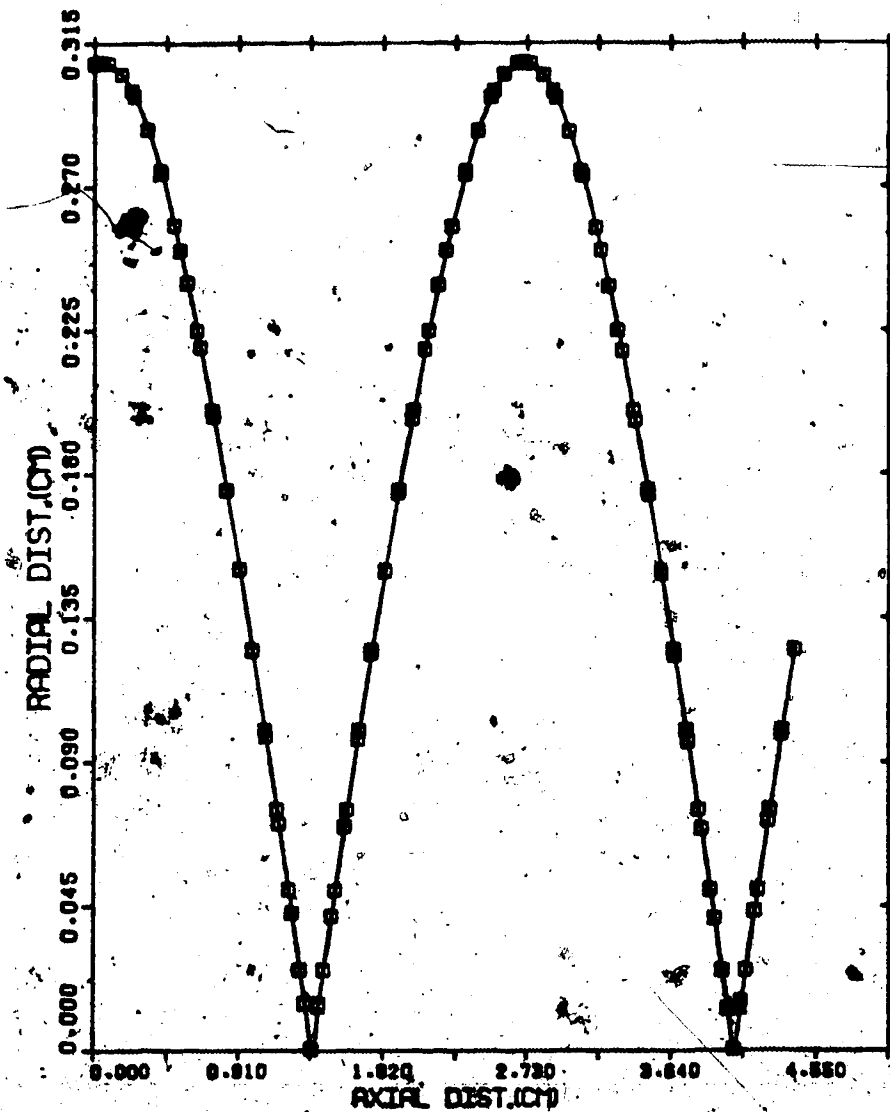


Figure 6.15 Ray path within region 2 (with initial outward radial velocity and initial position close to region 1).

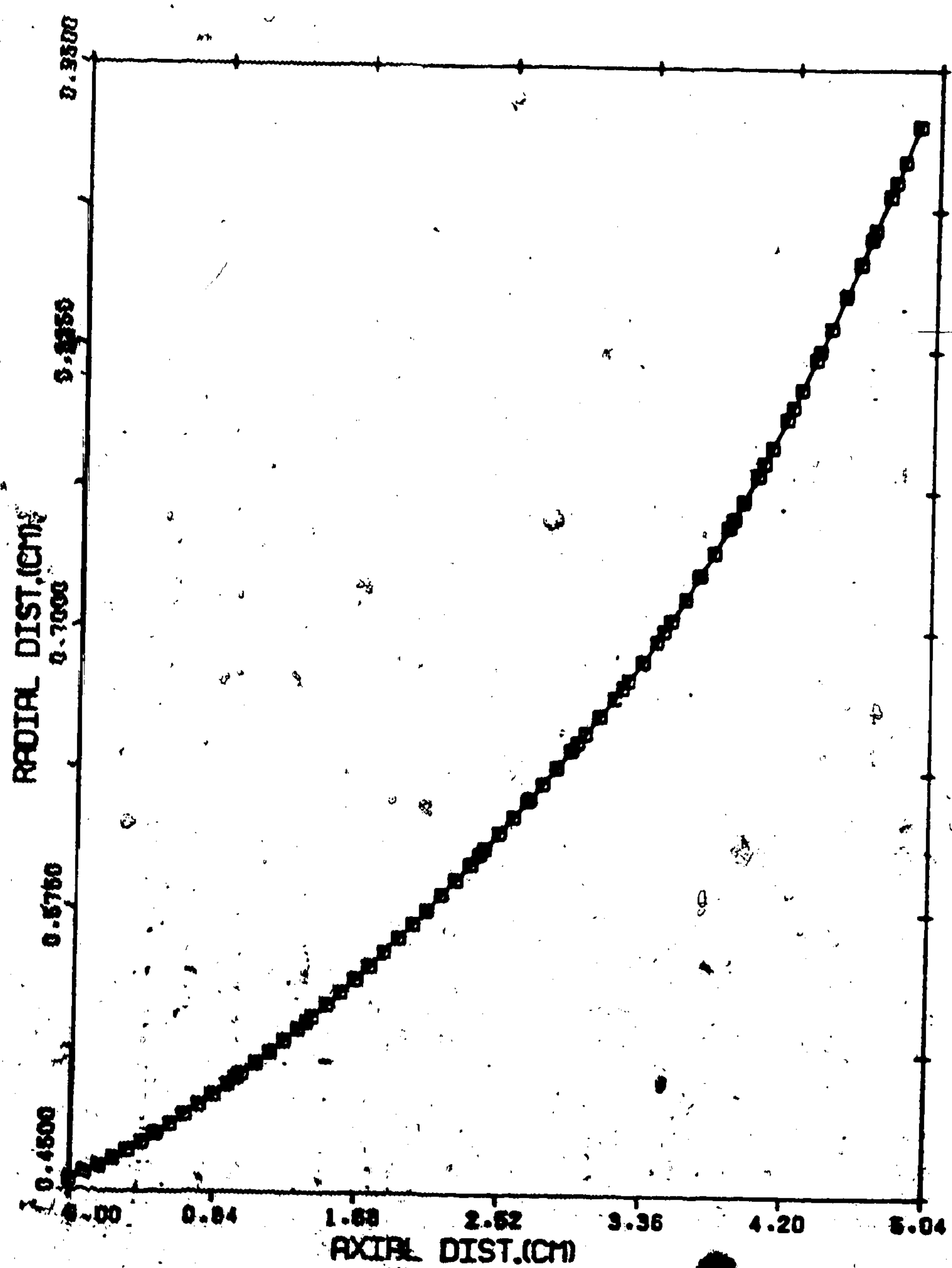


Figure 6.16 Ray path within region 3 (with initial outward radial velocity).

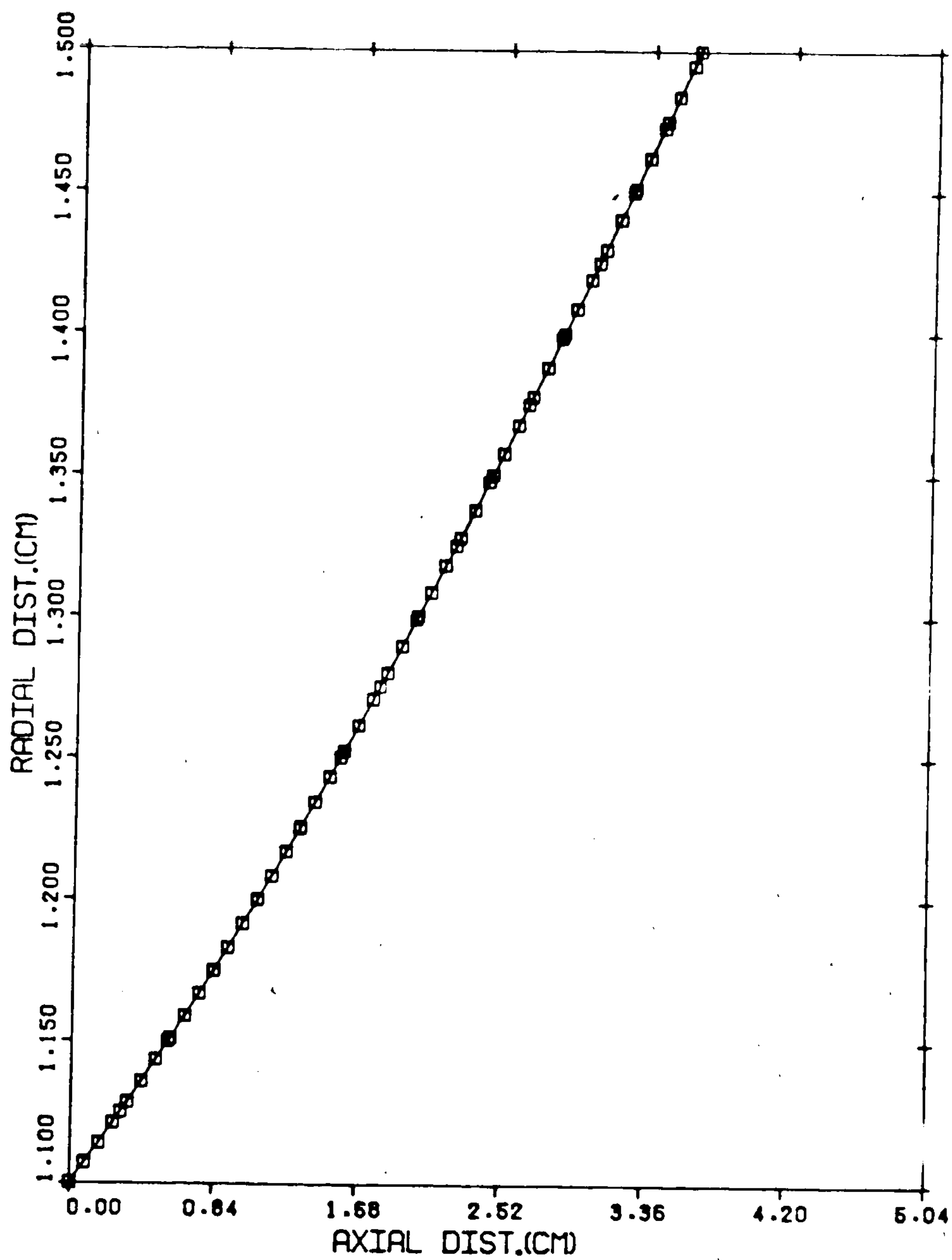


Figure 6.17 Ray path within region 4 (with initial outward radial velocity)

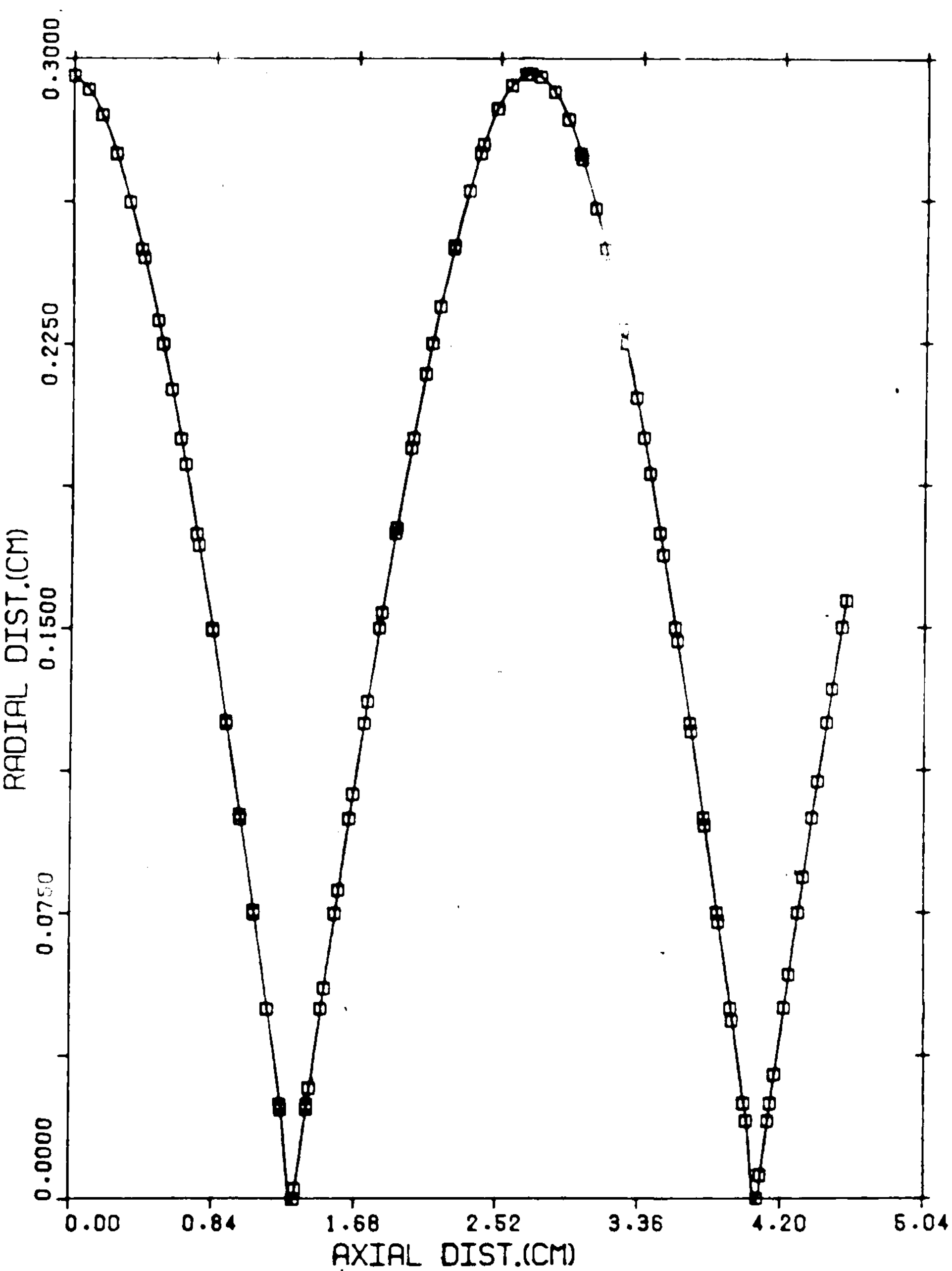


Figure 6.18 Ray path within region 1 (with initial inward radial velocity).

oscillation is found to be 5.48cm. A comparison to the period derived from eq (3.3.1.2), namely,

$$z_{period} = \frac{2\pi}{\lambda}\sqrt{1-\frac{N_0}{N_c}}\; v_z$$

where Ω is $\sqrt{\frac{c^2 N_0}{a_0^2 N_c}}$; v_z is 2.9×10^{10}cm/sec.; N_0 is $0.997 \times 10^{19}/cm^3$; a_0 is 0.337; N_c is $9.94 \times 10^{19}/cm^3$ (for $\lambda = 10.6\mu m$) and c is the speed of light, shows a 13% difference. This deviation arises from the choice of density profiles in the corresponding region. In this case, the plasma density in the region into which the ray enters is approximated by a non-parabolic increasing density profile. However, the density is assumed to vary according to a parabolic increasing density profile. This mismatch of density profile leads to the above deviation.

In fig. 6.19, a ray which enters the plasma core from region 2(see fig. 6.11) is displayed. The initial locations and directions of the ray are x=-0.43cm, y=-0.13cm, u_x=0.2868 x 10^{-1}, u_y=0.8241 x 10^{-2}. The ray approaches the inner core region gradually , reaches a minimum radial position and then exits the column. This phenomenon is indicated by a change in radial distance of the ray.

The behaviour of the rays when they enter into region 3 and region 4 (see fig. 6.11) are revealed in fig. 6.20 and fig. 6.21. The initial locations and directions for the ray in fig. 6.20 are x=0.404cm, y=0.513cm, u_x=-0.027, u_y=-0.034 and those in fig. 6.21 are x=-0.831cm, y=0.731cm, u_x=0.055, u_y=-0.049. As the rays approach the column axis, they enter into a medium of which the refractive index gradually decreases. Eventually, the rays are totally reflected off the column.

6.4 Absorbed energy and ponderomotive forces

In this section, the distribution of rays along the column is presented in terms of their locations at various transverse planes. The magnitudes of the absorbed energy and ponderomotive forces for a beam simulated with 10 rays are presented in terms of three dimensional plots.

Distributions of rays in the transverse planes located at the axial distances, z=0.0cm, 1.25cm, 2.5cm, 3.75cm and 5.0cm from the left end of the plasma column are

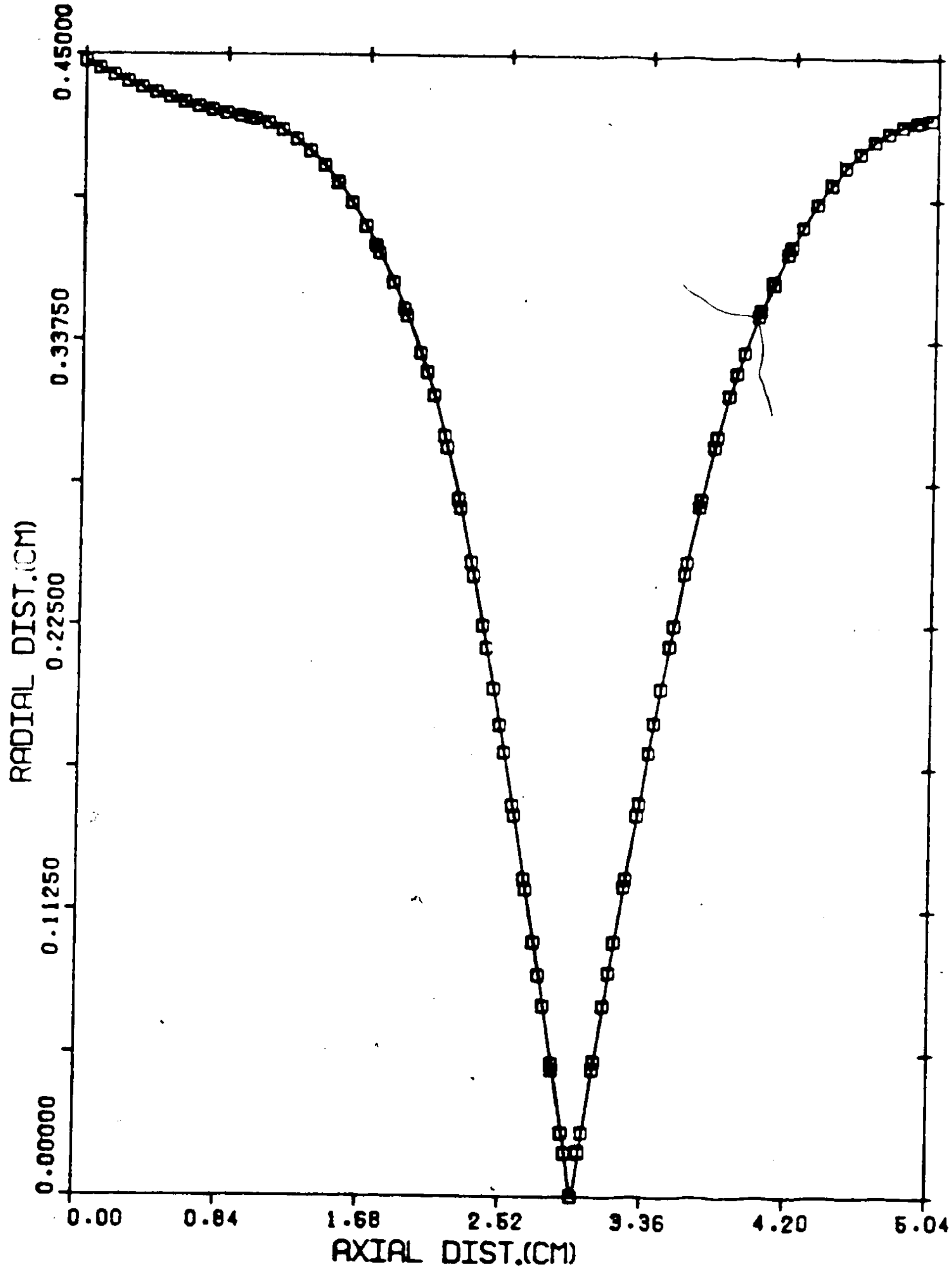


Figure 6.19 Ray path within region 2 (with initial inward radial velocity).

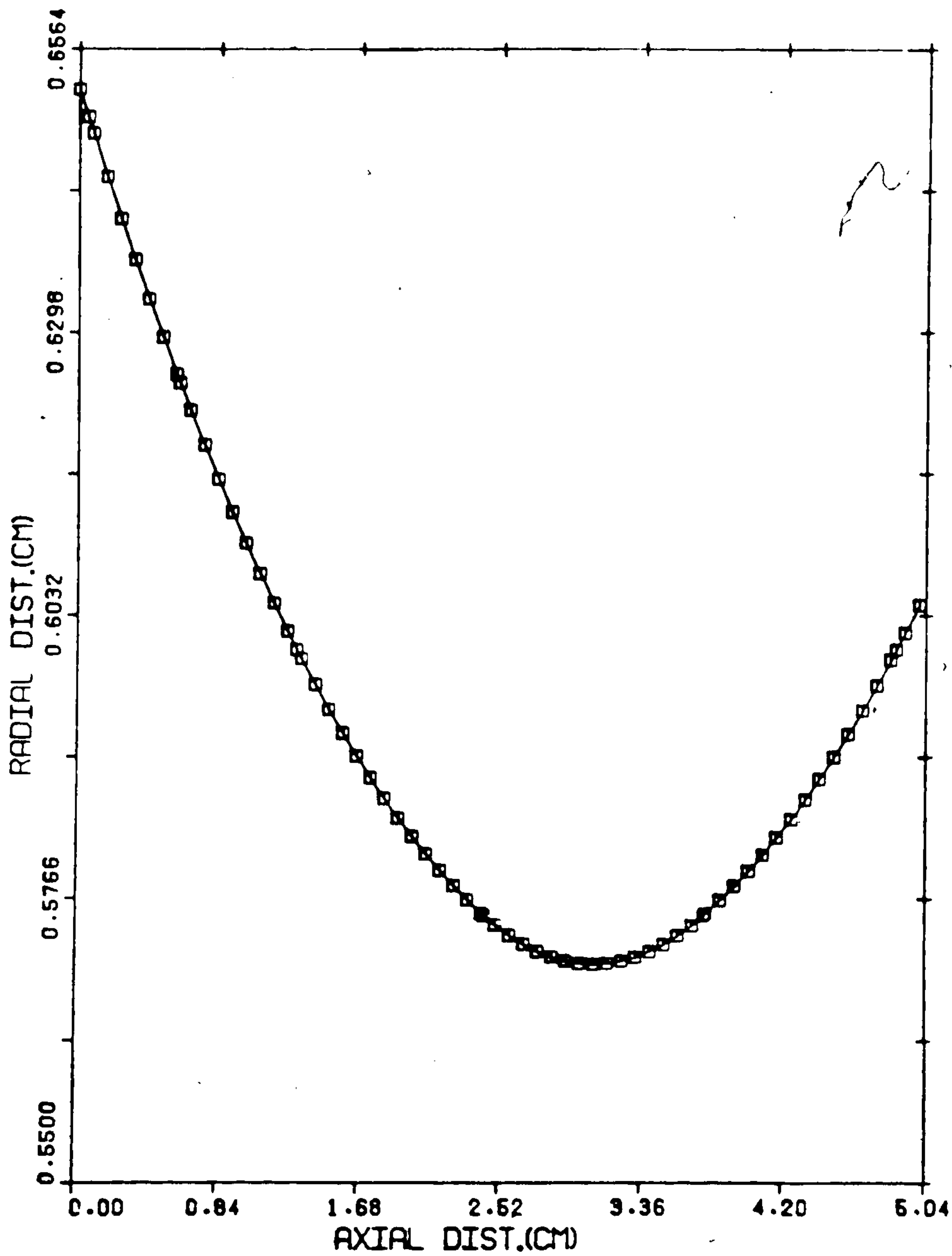


Figure 6.20 Ray path within region 3 (with initial inward radial velocity).

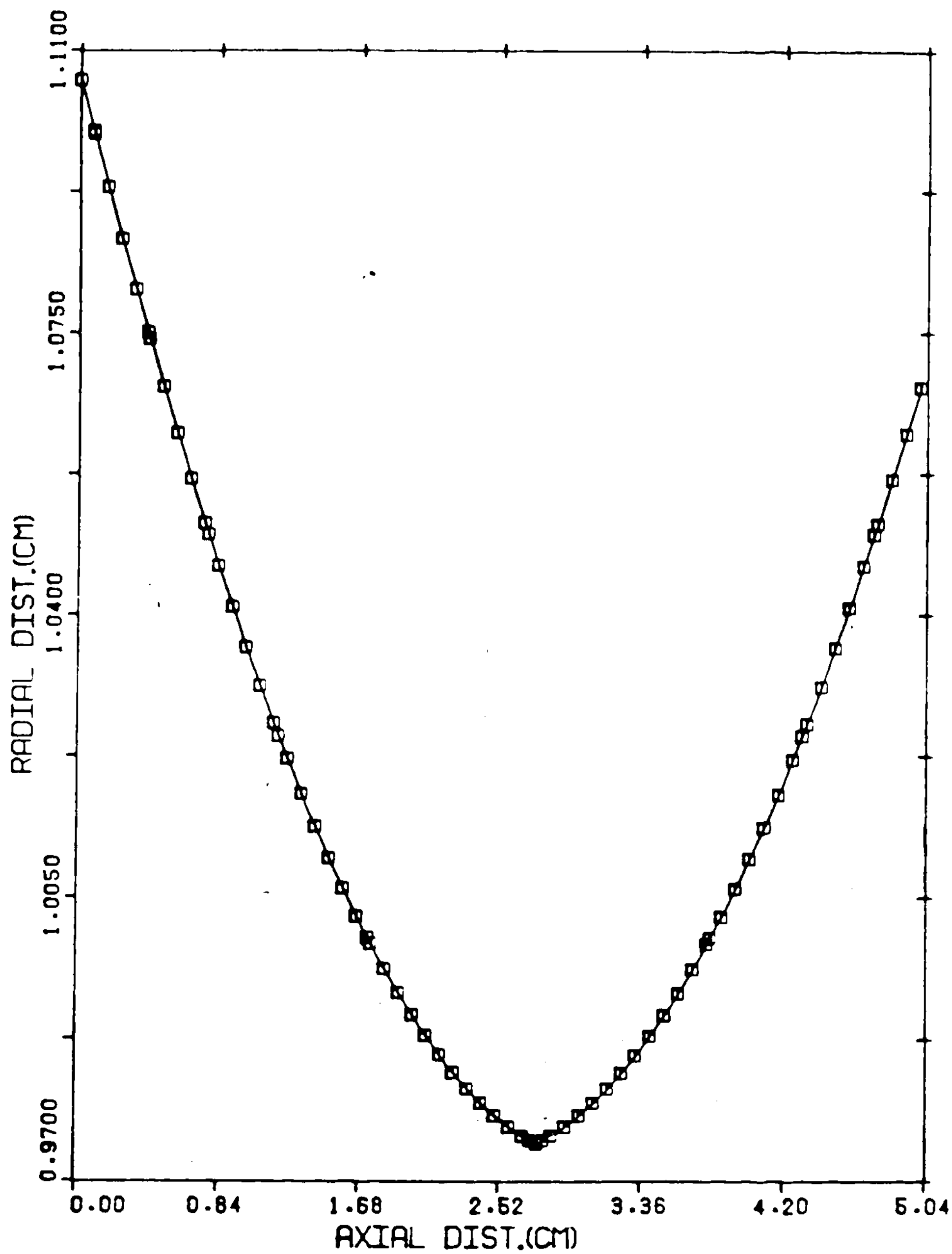


Figure 6.21 Ray path within region 4 (with initial inward radial velocity).

shown in figs 6.22 to 6.26 Cases for converging rays (figs 6.22a–6.26a) and diverging rays (fig 6.22b–6.26b) are compared From figs 6.22a to 6.26a, the distribution of ray locations indicates that the rays are focused and defocused as they propagate along the column At z=1.25cm and 5.0cm, almost all rays propagate within the first shell in the column, showing that the beam is focused at these locations.

From figs 6.22b to 6.26b, the behaviour of the rays is seen to be very similar to the set of rays displayed in figs 6.22a to 6.26a However, more rays are also seen to spread over the periphery of the column. This is a consequence of the divergence of the rays which implies that a higher proportion of the rays is distributed away from the column axis

The absorbed energy and ponderomotive forces associated with rays focused at the centre of the column are illustrated in figs 6.27 to 6.29. In fig 6.27, the absorbed energy per grid cell peaks at z=1.25cm and z=4.4cm. This maximum absorption is only a consequence of the rays concentrating at those locations and does not imply that strong absorption occurs in those regions. Moreover, the amount of absorbed energy in a cell at z=1.25cm is just slightly higher than that at z=4.4cm. This is due to the small and approximately equal magnitude of the absorption coefficients at both locations When the beam reaches the location at z=1.25cm, the beam power is not strongly absorbed and most of the power is transmitted down the column. At z=4.4cm, the input beam power does not decrease significantly. Moreover, due to the approximately equal absorption coefficient, the absorbed power is about the same.

Plots of radial and axial ponderomotive forces along the column are shown in fig 6.28 and fig. 6.29. Both forces have maximum magnitudes at the region with the highest radiation intensity. Negative amplitudes imply that the forces and radial displacement are in opposite directions. With an input laser intensity of 1.0×10^{9}watts, the maximum magnitude of the radial and axial ponderomotive forces are found to be 0.198×10^{4} (dynes/cm^{3}) and the axial ponderomotive force to be 0.54×10^{-3} (dynes/cm^{3}) respectively. By comparing these values to the hydrodynamic force(for T=100eV, $dN/dr=8.8 \times 10^{17}$ /cm^{4}, over a scale length of 0.05cm, $k=1.6 \times 10^{-12}$erg/eV, $dP/dr=kT(dN/dr)=1.41 \times 10^{8}$ dynes/cm^{3}), the ponderomotive forces are far smaller than the hydrodynamic ones. These force components will become significant when the laser intensity becomes much higher

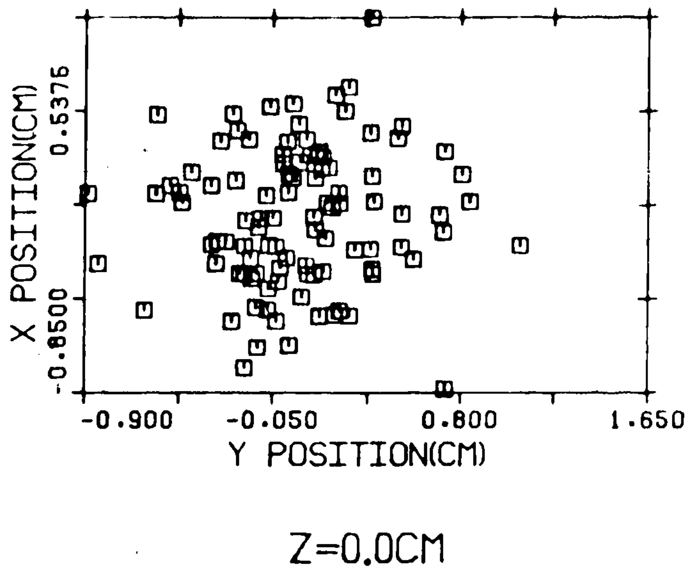


Figure 6.22a Ray distribution with the front end of the plasma column placed at 135cm from lens. The focal length of the lens is assumed to be 150cm.

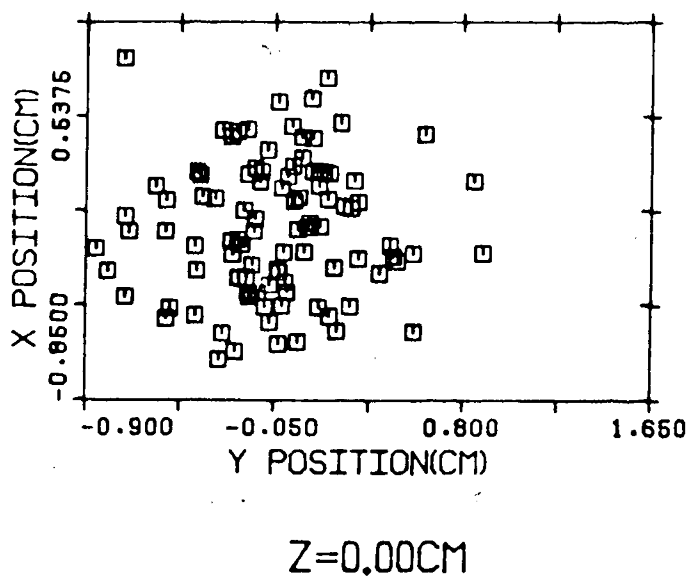


Figure 6.22b Ray distribution with the front end of the plasma column placed at 165cm from lens. The focal length of the lens is assumed to be 150cm.

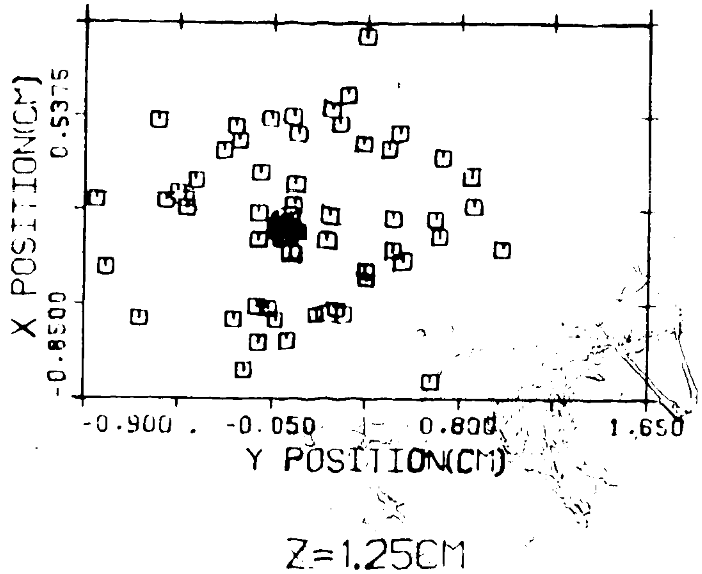


Figure 6.23a Ray distribution with the front end of the plasma column placed at 135cm from lens. The focal length of the lens is assumed to be 150cm.

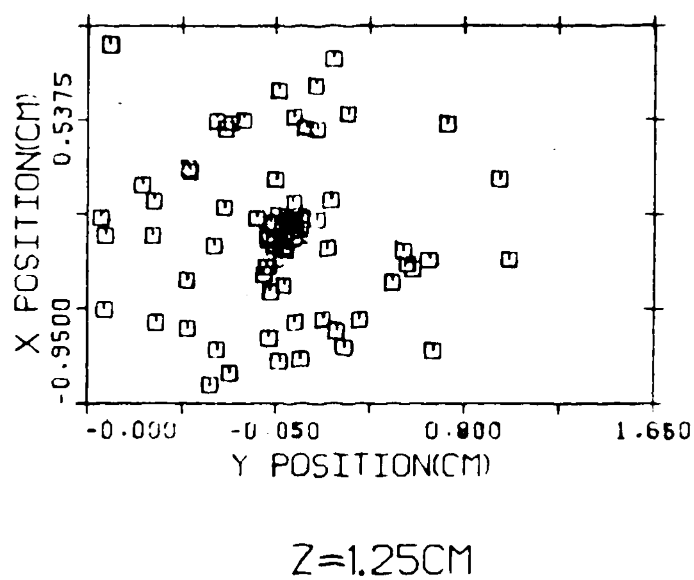


Figure 6.23b Ray distribution with the front end of the plasma column placed at 165cm from lens. The focal length of the lens is assumed to be 150cm.

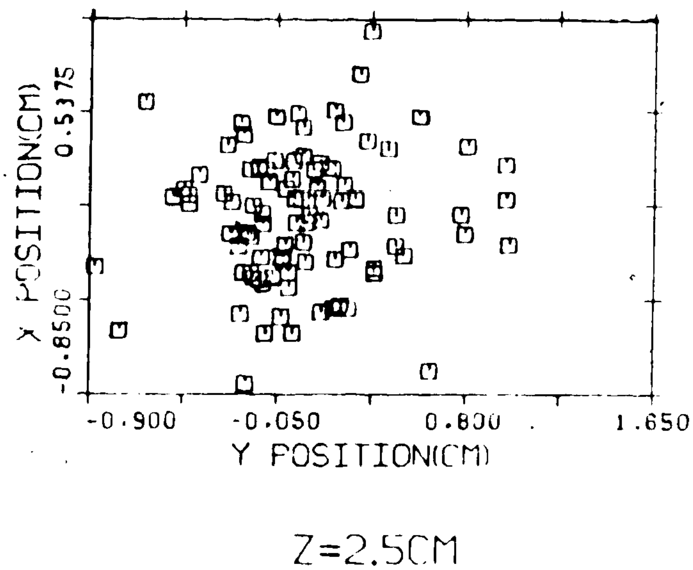


Figure 6.24a Ray distribution with the front end of the plasma column placed at 135cm from lens The focal length of the lens is assumed to be 150cm

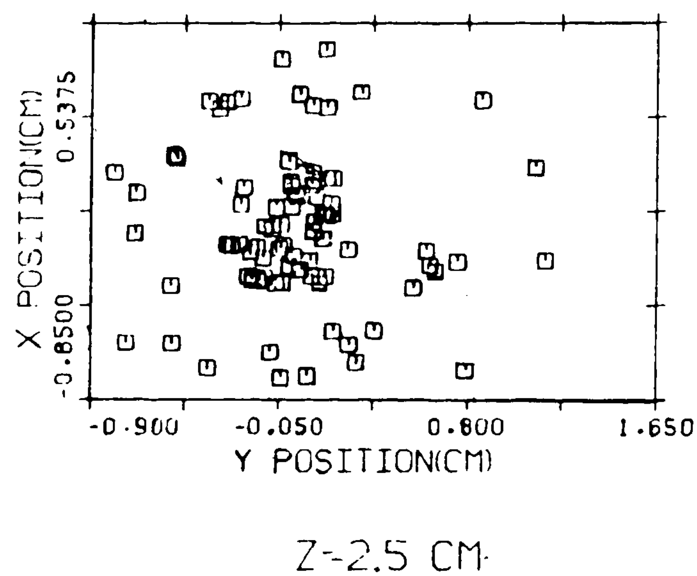


Figure 6.24b Ray distribution with the front end of the plasma column placed at 165cm from lens. The focal length of the lens is assumed to be 150cm.

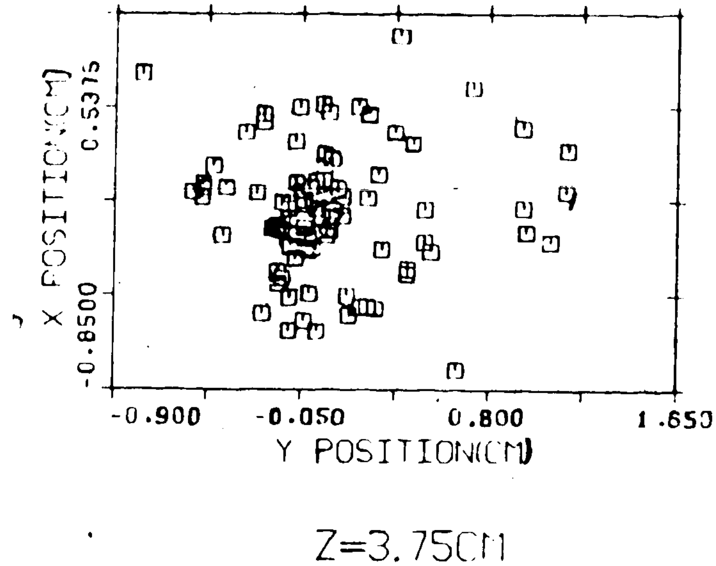


Figure 6.25a Ray distribution with the front end of the plasma column placed at 135cm from lens ~~The focal~~ length of the lens is assumed to be 150cm.

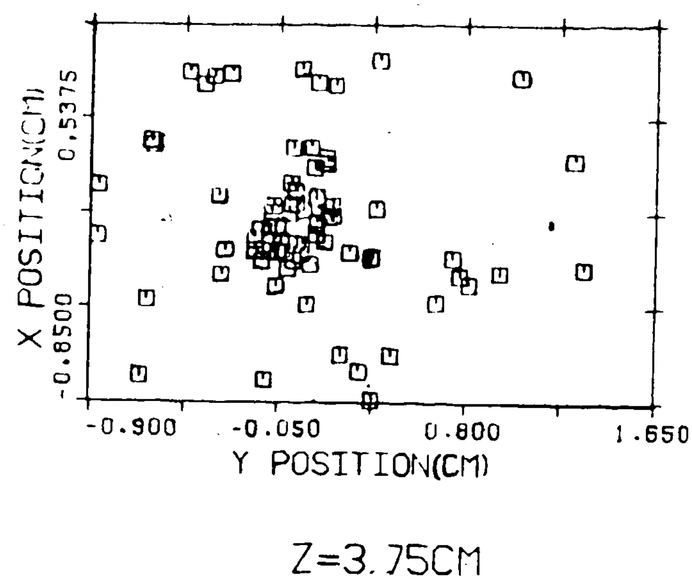
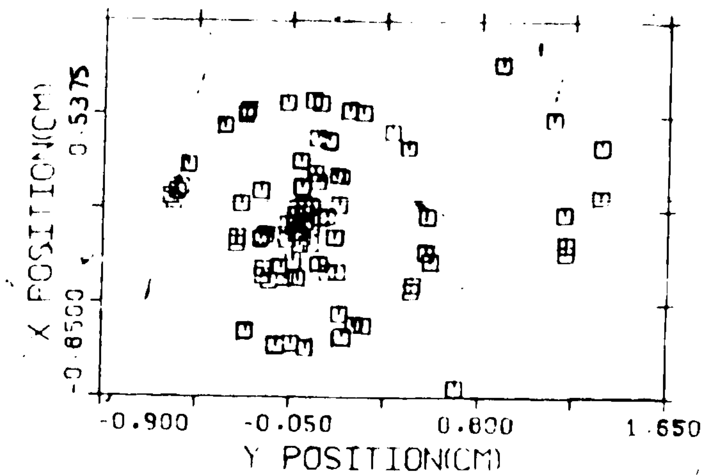
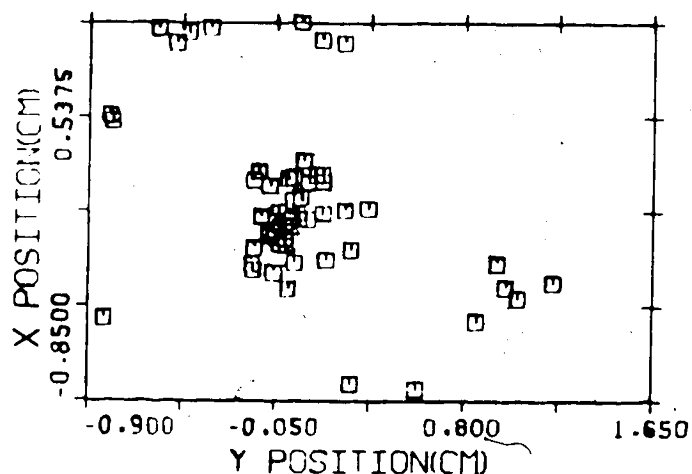


Figure 6.25b Ray distribution with the front end of the plasma column placed at 165cm from lens. The focal length of the lens is assumed to be 150cm.



Z=5.0CM

Figure 6.26a Ray distribution with the front end of the plasma column placed at 135cm from lens. The focal length of the lens is assumed to be 150cm.



Z=5.00CM

Figure 6.26 b Ray distribution with the front end of the plasma column placed at 165cm from lens. The focal length of the lens is assumed to be 150cm.

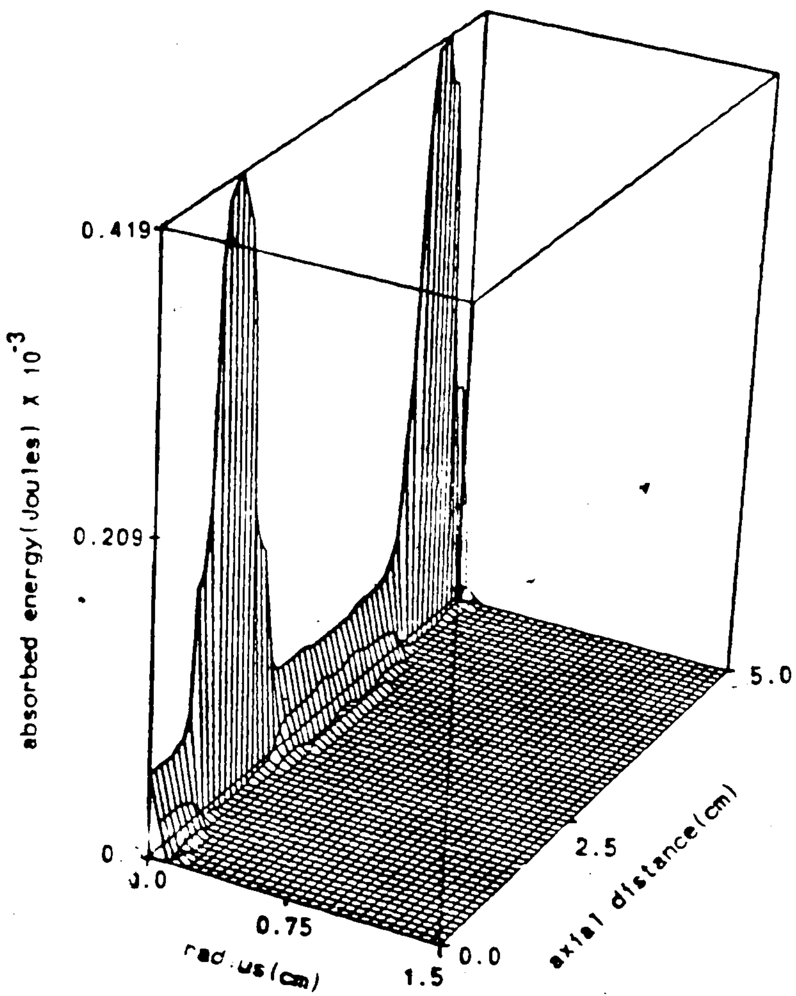


Figure 6.27 Distribution of absorbed energy within plasma column (with beam focused at the middle of the column).

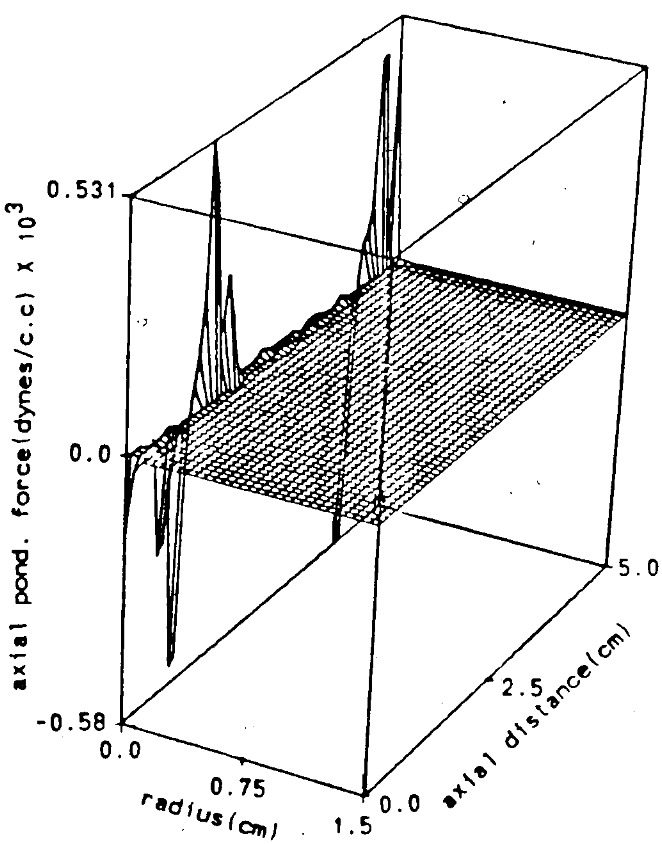


Figure 6.28 Distribution of axial ponderomotive force within the column.

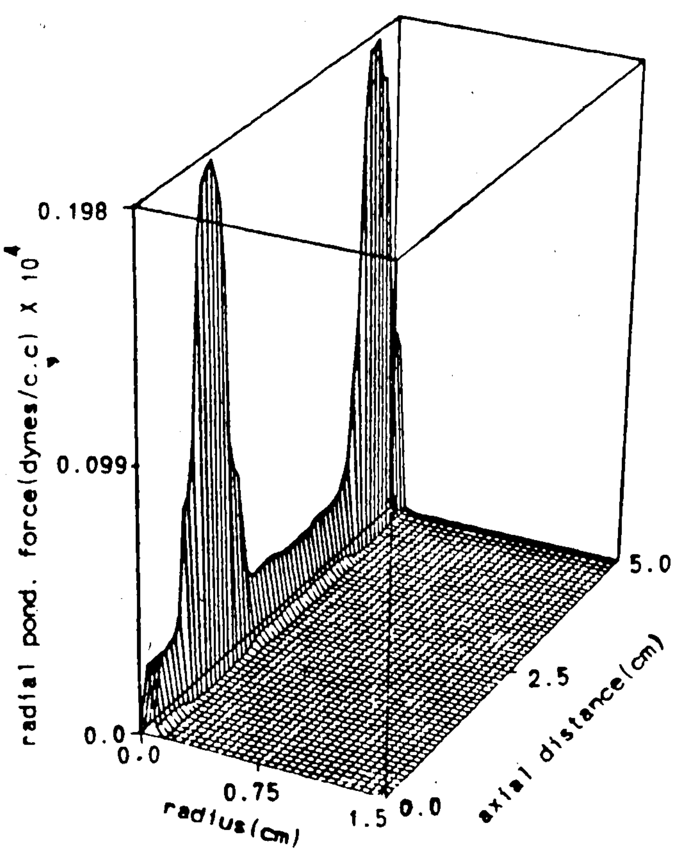


Figure 6.29 Distribution of radial ponderomotive force within the column.

Conclusion

The propagation of a focused laser beam in a vacuum and in a plasma medium is investigated by using a ray tracing technique which accounts for the diffraction effects of the beam. The effect is introduced by means of a phase-space distribution function which was suggested by Tappert. The distribution function provides a range of values for the ray directions so that rays are not focused at one point, but are spread out at the focal region. If the laser beam intensity assumes a Gaussian profile at the lens plane, rays will be then distributed across the focal plane in a Gaussian manner. Results show that this method can generate rays which give the diffraction limited focal spot size. Optical defects such as aberration, are modelled by introducing an incoherence factor into the distribution function. By means of this factor, the focal spot size can be adjusted to meet experimental measurements.

Rays are traced through the plasma by means of analytic methods. The ray equation is solved for different kinds of refractive index profiles. Ray trajectories within the medium are traced in terms of these solutions. Since the plasma density is spatially non-homogeneous, appropriate density profiles are used to describe the density in various regions. Corresponding ray path solutions are used. A testing density profile with an on-axis minimum obtained from an MHD simulation of a laser heated plasma confined by a solenoidal magnetic field is used as a sample run for the ray tracing package. Results indicate that rays are trapped within regions with a parabolic radial density variation; rays are reflected off axis when they propagate in regions having a radially decreasing density profile. Focusing and defocusing effects of the beam within the predefined profile are shown by the distribution of ray locations at various axial planes.

The plasma density profile used in the simulation is assumed to be axially independent. However, in actual plasma density distribution evaluated by the shell code, the density varies axially as well. In the ray tracing routine, this axial density change is taken into account by the appropriate fitting of density profiles over the grid cells. Since the choice of density profiles is determined by the density values in the grid cells, any variations in density values will give different choices for the density profile.

An extension of the ray tracing technique to gas target experiments will be of interest and use for future research. For the gas targets, there are no magnetic fields confining the plasma and as a result, the plasma expands freely in all directions upon heating The rays will intersect the plasma boundary at different locations in subsequent times. Moreover, because of the high plasma density, rays can be scattered as they reach the region with critical densities. The problems of a moving plasma boundary and beam scattering are significant factors to be considered in the ray tracing program for gas target experiments.

Throughout the analysis, the spatial distribution of the laser beam is assumed to be composed of the fundamental Gaussian mode. In a real situation, the beam may have a different mode structure mixed together(see footnote) and cylindrical symmetry no longer holds. The intensity of the beam is expected to vary with angular and radial positions. Thus, the choice of ray locations and directions are determined by the radial and azimuthal co-ordinates. Consequently, the number of rays required for simulation will be increased significantly and will increase the operating cost of the simulation program considerably. Further investigation on this problem is needed.

Diagrams of different mode structure are given in fig. 4.12 in 'An introduction to optical electronics' by A. Yariv

Bibliography

Chapter 1

(1) J. Nuckolls, L. Wood, A. Thiessen, and G. Zimmerman, Nature, 239 (Sep. 1972)

(2) J.M. Dawson, R.E. Kidder, A. Hertzberg, G.C. Vlases and H.G. Ahlstrom, IAEA Fourth Conference on Plasma Physics and Controlled Nuclear Fusion Research, Madison, Wisconsin (1971).

(3) S. Humphries, Jr., Plasma Phys. 16, (1974) 623.

(4) S.A. Mani, J.E. Eninger, J. Wallace, Nucl. Fusion 15 (1975) 371.

(5) M.D. Feit, J.A. Fleck, Jr., Appl. Phys. Lett., 28 (1976) 121.

(6) M.D. Feit, D.E. Maiden, Appl. Phys. Lett 28 (1976) 331.

(7) M.D. Feit, J.A. Fleck, Jr., Appl. Phys. Lett., 29 (1976) 234.

(8) M.D. Feit, J.A. Fleck, Jr., and J.R. Morris, J. Appl. Phys., 48 (1977) 3301.

(9) J.N. McMullin, C.E. Capjack, C.R. James, Phys. Fluids 21 (1978) 1828; Phys. Fluids. 22 (1979) 953.

(10) L.C. Steinhauer, H.G. Ahlstrom, Phys. Fluids 14 (1971) 1109.

(11) L.C. Steinhauer, Phys. Fluids 19 (1976) 1740.

(12) A.A. Offenberger, M.R. Cervenan, P.R. Smy, J. Appl. Phys. 47 (1976) 494.

(13) C. Cohn, W. Halverson, Phys. Lett., 49A (1974) 95.

(14) G.C. Vlases, N.A. Amherd, Appl. Phys. Lett 24 (1974) 93.

(15) H.D. Dudder, D.B. Henderson, Phys. Comm. 10 (1975) 155.

(16) M. Hubbard, A. Montes, Culham Laboratory Report, CLM-R189 (Jul. 1978).

(17) G. Rinker, G. Bohannon, IEEE Trans. on Plasma Science PS-8 (1980) 55.

(18) F. Tappert, J. Opt Soc. Am. 66 (1976) 1368.

(19) J.N. McMullin, R.D. Milroy, C.E. Capjack, Phys. Fluids 22 (1979) 1913.

(20) F. Chen, in Laser Interaction and Related Plasma Phenomena 3A (Schwartz, Hora, Eds) Plenum Press (1971) 291.

Chapter 2

(1) Born, Principle of Optics, Pergamon Press (1980) 435.

(2) E. Wigner, Phy. Rev. 40 (1932) 479.

(3) A. Papoulis, J. Opt Soc. Am. 64 (1974) 779.

(4) F. Tappert, J.Opt.Soc.Am. 66 (1976) 1368

(5) Siegman, Introduction to Lasers and Masers, Ch. 8, McGraw-Hill (1971).

(6) Papoulis, Probability, Random Variables and Stochastic Processes, McGraw-Hill (1965).

(7) H.D. Dudder, D.B. Henderson, Phys. Comm. 10 (1975) 155.

(8) Y.W. Lee, Statistical Theory of Communication, Ch.7, John Wiley & Sons., Inc. (1960)

Chapter 3

(1) E.G. Harris, Phy. Rev. 138 (1965) B479.

(2) S. Weinberg, Phy. Rev. 126 (1962) 1899.

(3) Stix, The Theory of Plasma Waves, McGraw-Hill (1962) 56.

(4) G.C. Pomraning, "High temperature radiative transfer and hydrodynamics", in Progress in High Temperature Physics and Chemistry, 4 (C.A. Rouse, Eds) Pergamon Press, New York (1971) 39.

(5) J. Nuckolls, L. Wood, A. Thiessen, and G. Zimmerman, Nature 239 (1972) 239.

(6) S.A. Mani, J.E. Eninger, J. Wallace, Nucl. Fusion 15 (1975) 371.

(7) L.C. Steinhauer, H.G. Ahlstrom, Phys. Fluids 14 (1971) 1109.

Chapter 4

(1) Johnston and Dawson, Phys. Fluids 16 (1973) 722.

(2) F. Chen in Laser Interaction and Related Plasma Phenomena 3A (Schwartz, Hora, Eds) Plenum Press (1971) 291.

Appendix 1

1.1 Derivation of Poynting vector in eq.(2.1.8)

Assume that the electric field vector $\vec{E}$ has components $\vec{E}_x, \vec{E}_z$ where $|\vec{E}_z| << |\vec{E}_x|$

$$\vec{E} = E_x\hat{x} + E_z\hat{z}$$

From Maxwell's equation,

$$\vec{\nabla}\cdot\vec{E} = 0$$

we get

$$\frac{\partial E_x}{\partial x} + \frac{\partial E_z}{\partial z} = 0 \qquad (1)$$

$$\frac{\partial E_x}{\partial x} = \frac{-\partial E_z}{\partial z}$$

Let
$$E_x = \varepsilon(x,y,z)e^{i(kz-\omega t)} \qquad (2)$$

Therefore, equation(1) becomes

$$\frac{\partial E_z}{\partial z} = -\frac{\partial \varepsilon}{\partial x}e^{i(kz-\omega t)}$$

$$E_z \doteq \frac{i}{k}\frac{\partial \varepsilon}{\partial x}e^{i(kz-\omega t)} \qquad (3)$$

Again from Maxwell's equation,

$$\vec{\nabla}\times\vec{E} = -\frac{1}{c}\frac{\partial \vec{B}}{\partial t}$$

With

$$\vec{B} = \vec{B}(x,y,z)e^{i(kz-\omega t)}$$

$$\vec{B} = \frac{-ic}{\omega}\vec{\nabla} \times \vec{E}$$

Substituting $\vec{E}$ with eq.(2) and (3),

$$\vec{B} = \frac{-ic}{\omega}\left[\frac{\partial \mathcal{E}_z}{\partial y}\hat{x} + \left(\frac{\partial \mathcal{E}_x}{\partial z} - \frac{\partial \mathcal{E}_z}{\partial x}\right)\hat{y} - \frac{\partial \mathcal{E}_x}{\partial y}\hat{z}\right]$$

$$\doteq \frac{-ic}{\omega}\left[\frac{i}{k}\frac{\partial^2 \varepsilon}{\partial x\, \partial y}\hat{x} + \left(ik\varepsilon + \frac{\partial \varepsilon}{\partial z} - \frac{i}{k}\frac{\partial^2 \varepsilon}{\partial x^2}\right)\hat{y} - \frac{\partial \varepsilon}{\partial y}\hat{z}\right]e^{i(kz-\omega t)}$$

Dropping the second order derivative of ε,

$$\vec{B} = \frac{-ic}{\omega}\left[\left(ik\varepsilon + \frac{\partial \varepsilon}{\partial z}\right)\hat{y} - \frac{\partial \varepsilon}{\partial y}\hat{z}\right]e^{i(kz-\omega t)}$$

$$= \left[\varepsilon\hat{y} + \frac{ic}{\omega}\left(-\frac{\partial \varepsilon}{\partial z}\hat{y} + \frac{\partial \varepsilon}{\partial y}\hat{z}\right)\right]e^{i(kz-\omega t)}$$

The Poynting vector $\vec{S}$ is

$$\vec{S} = \frac{c}{8\pi}\,\mathrm{Re}[\vec{E} \times \vec{B}^*]$$

$$= \frac{c}{8\pi}\,\mathrm{Re}\left[\left(\varepsilon\hat{x} + \frac{i}{k}\frac{\partial \varepsilon}{\partial x}\hat{z}\right) \times \left[\varepsilon^*\hat{y} - \frac{ic}{\omega}\left(\frac{-\partial \varepsilon^*}{\partial z}\hat{y} + \frac{\partial \varepsilon^*}{\partial y}\hat{z}\right)\right]\right]$$

$$= \frac{c}{8\pi}\,\mathrm{Re}\left[|\varepsilon|^2\hat{z} - \frac{i}{k}\varepsilon^*\frac{\partial \varepsilon}{\partial x}\hat{x} + \frac{ic}{\omega}\varepsilon\frac{\partial \varepsilon^*}{\partial z}\hat{z} + \frac{ic}{k\omega}\frac{\partial \varepsilon}{\partial x}\frac{\partial \varepsilon^*}{\partial z}\hat{x} + \frac{ic}{\omega}\varepsilon\frac{\partial \varepsilon^*}{\partial y}\hat{y}\right]$$

Assume

$$|\varepsilon|^2 >> |\frac{1}{k}\varepsilon\frac{\partial\varepsilon^*}{\partial z}| \qquad |\varepsilon^*\frac{\partial\varepsilon}{\partial x}| >> \frac{1}{k}|\frac{\partial\varepsilon}{\partial x}||\frac{\partial\varepsilon^*}{\partial z}|$$

Then,

$$\vec{S} = \frac{c}{8\pi} \mathrm{Re}[|\varepsilon|^2 \hat{z} - \frac{i}{k}\varepsilon^*\frac{\partial\varepsilon}{\partial x} \hat{x} + \frac{ic}{\omega}\varepsilon\frac{\partial\varepsilon^*}{\partial y} \hat{y}]$$

$$= \frac{c}{8\pi} [|\varepsilon|^2\hat{z} + \frac{i}{2k}[(\varepsilon\frac{\partial\varepsilon^*}{\partial x} - \varepsilon^*\frac{\partial\varepsilon}{\partial x}) \hat{x} + (\varepsilon\frac{\partial\varepsilon^*}{\partial y} - \varepsilon^*\frac{\partial\varepsilon}{\partial y}) \hat{y}]$$

$$= \frac{c}{8\pi} [|\varepsilon|^2\hat{z} + \frac{i}{2k}[\varepsilon\frac{\partial\varepsilon^*}{\partial x} \hat{x} + \varepsilon\frac{\partial\varepsilon^*}{\partial y} \hat{y} - (\varepsilon^*\frac{\partial\varepsilon}{\partial x} \hat{x} + \varepsilon^*\frac{\partial\varepsilon}{\partial y} \hat{y})]]$$

$$= \frac{c}{8\pi} [|\varepsilon|^2\hat{z} + \frac{i}{2k}(\varepsilon\vec{\nabla}_\perp\varepsilon^* - \varepsilon^*\vec{\nabla}_\perp\varepsilon)]$$

$$= \frac{c}{8\pi} [|\varepsilon|^2\hat{z} - \frac{i}{2k}(\varepsilon^*\vec{\nabla}_\perp\varepsilon - \varepsilon\vec{\nabla}_\perp\varepsilon^*)]$$

Appendix 2

2.1 Derivation of the electric field after the lens plane

The far field approximation of the electric field of a laser beam at a distance of z from the source is given as[5]

$$u(r,z) = \sqrt{\frac{2}{\pi}}\frac{1}{w(z)} e^{i(kz-\phi(z))} e^{-\frac{r^2}{w^2(z)}} e^{-\frac{ikr^2}{2R(z)}} \tag{1}$$

where

$$\phi = \tan^{-1}\left(\frac{\lambda z}{\pi w_0^2}\right)$$

$$R(z) = z\left[1 + \left(\frac{\pi w_0^2}{\lambda z}\right)^2\right]$$

$$w^2(z) = w_0^2\left[1 + \left(\frac{\lambda z}{\pi w_0^2}\right)^2\right]$$

$$r^2 = x^2 + y^2 \qquad \text{and} \qquad r << R$$

w_0 is the beam waist of the laser.

If a thin lens is placed at a distance z_0 from the origin, the field amplitude of the light beam is

$$u(r,z_0) = \sqrt{\frac{2}{\pi}}\frac{1}{w(z_0)} e^{-i(kz_0-\phi(z_0))} e^{-\frac{r^2}{w^2(z_0)}} e^{-\frac{ikr^2}{2R(z_0)}} \tag{2}$$

On passing through the lens, the field amplitude becomes

$$u(r,z_1) = \sqrt{\frac{2}{\pi}}\,\frac{1}{w(z_1)}\, e^{-i(kz_1-\Phi(z_1))}\, e^{-\frac{r^2}{w^2(z_1)}}\, e^{\frac{ikr^2}{2R(z_1)}} \tag{3}$$

where $R(z_1)$ is negative (by convention) since the beam converges. But the radius of curvature of the wavefront is related to the focal length of a thin lens by

$$\frac{1}{R(z_1)} = [\frac{1}{R(z_0)} - \frac{1}{f_L}] \tag{4}$$

The field becomes

$$u(r,z_1) = \sqrt{\frac{2}{\pi}}\,\frac{1}{w(z_1)}\, e^{-i(kz_1-\Phi(z_1))}\, e^{-r^2[\frac{1}{w^2(z_1)} - \frac{ik}{2R(z_0)} + \frac{ik}{2f_L}]} \tag{5}$$

Since the lens is assumed to be thin, the axial distance and spotsize are approximated as

$$z_0 = z_1, \quad w(z_0) = w(z_1)$$

The field becomes

$$u(r,z_1) = \sqrt{\frac{2}{\pi}}\,\frac{1}{w(z_0)}\, e^{-i(kz_0-\Phi(z_0))}\, e^{-r^2[\frac{1}{w^2(z_0)} + \frac{ik}{2f_L} - \frac{ik}{2z_0}]} \tag{6}$$

As $z_0 >> 1$ for far field,

$$= \sqrt{\frac{2}{\pi}}\,\frac{1}{w(z_0)}\, e^{-i(kz_0-\Phi(z_0))}\, e^{-r^2[\frac{1}{w^2(z_0)} + \frac{ik}{2f_L}]}$$

Let

$$E_0 = \sqrt{\frac{2}{\pi}} \frac{1}{w(z_0)} e^{-i(kz_0 - \phi(z_0))}$$

the electric field becomes

$$u(r\ ,z_1) = E_0\, e^{\frac{-r^2}{w^2(z_0)}} e^{\frac{-ikr}{2f_L}} \tag{7}$$

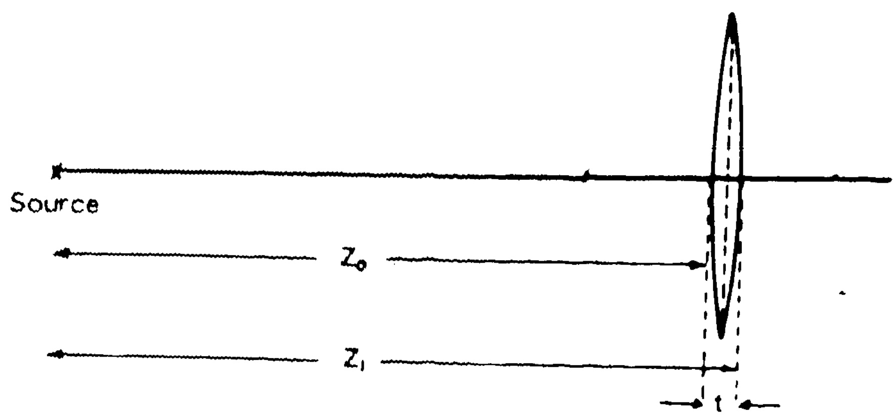


Fig 2.1 Optical system described in Appendix 2

Appendix 3

3.1 Program listings and flowcharts

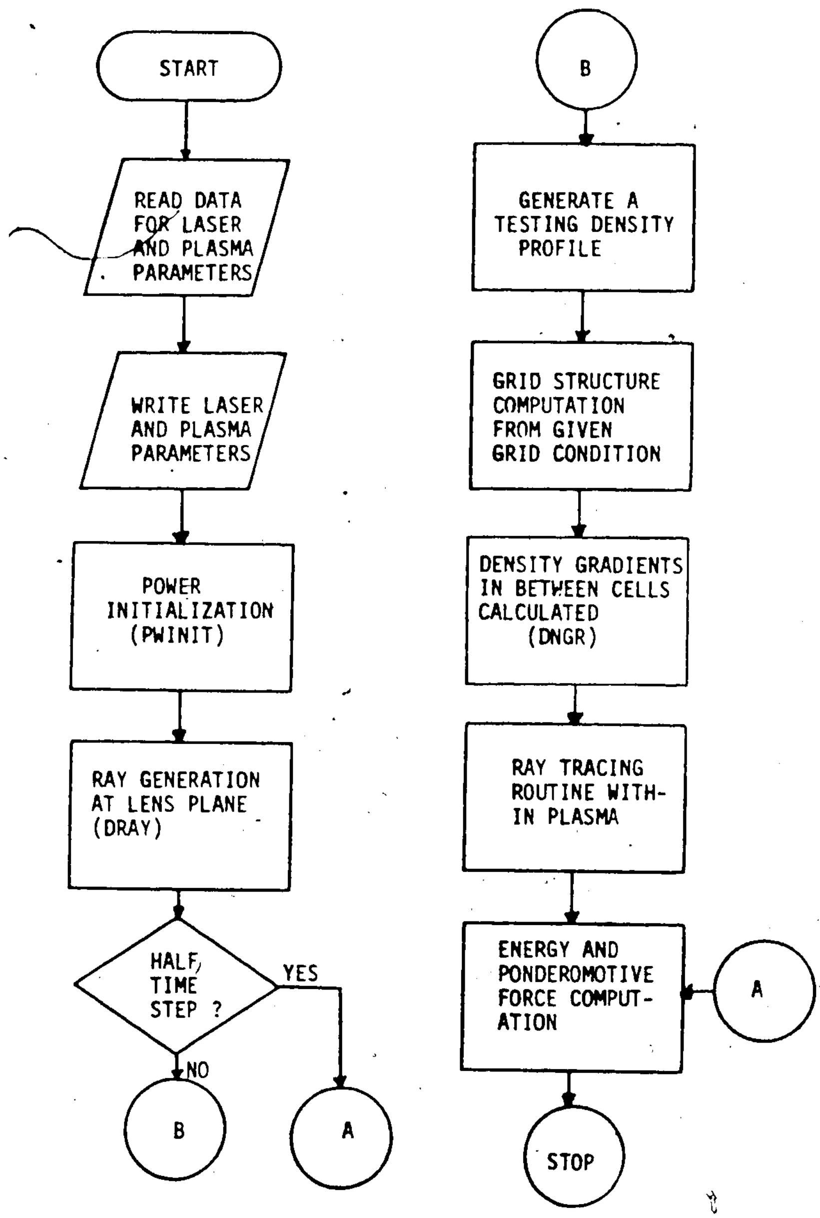


CHART I. PROGRAM FOR TESTING SUBROUTINE PACKAGES

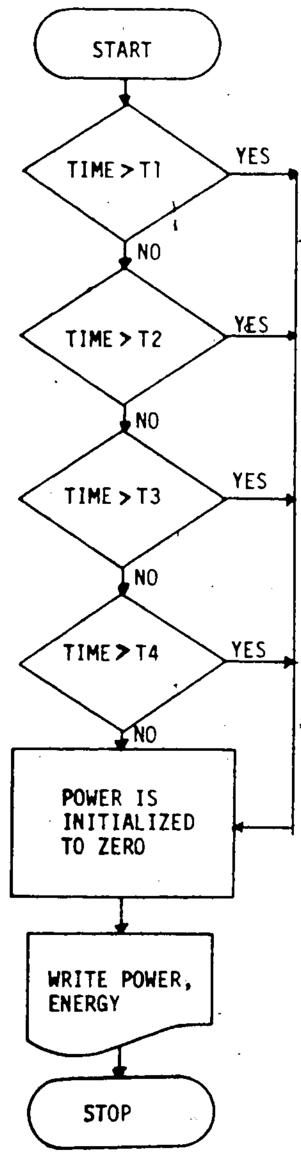


CHART 2. POWER INITIALIZATION ALGORITHM

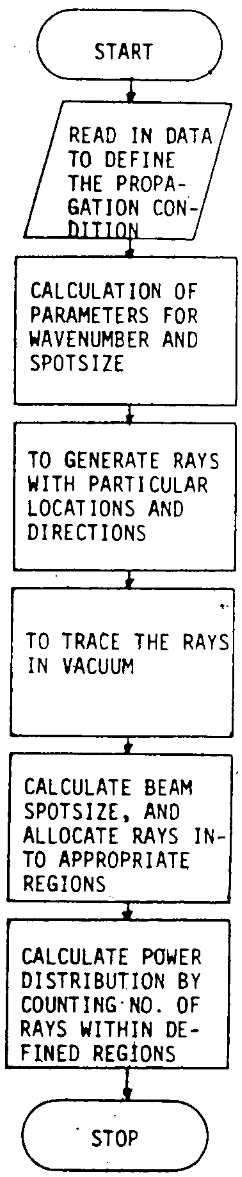


CHART 3. RAY GENERATION ALGORITHM

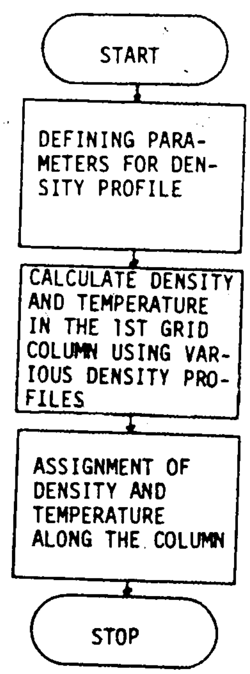


CHART 4. DENSITY AND TEMPERATURE PROFILE ALGORITHM

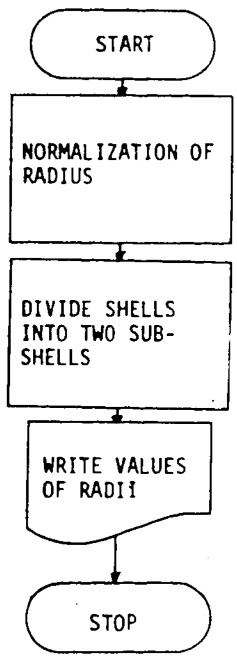


CHART 5. FINE GRID STRUCTURE ALGORITHM

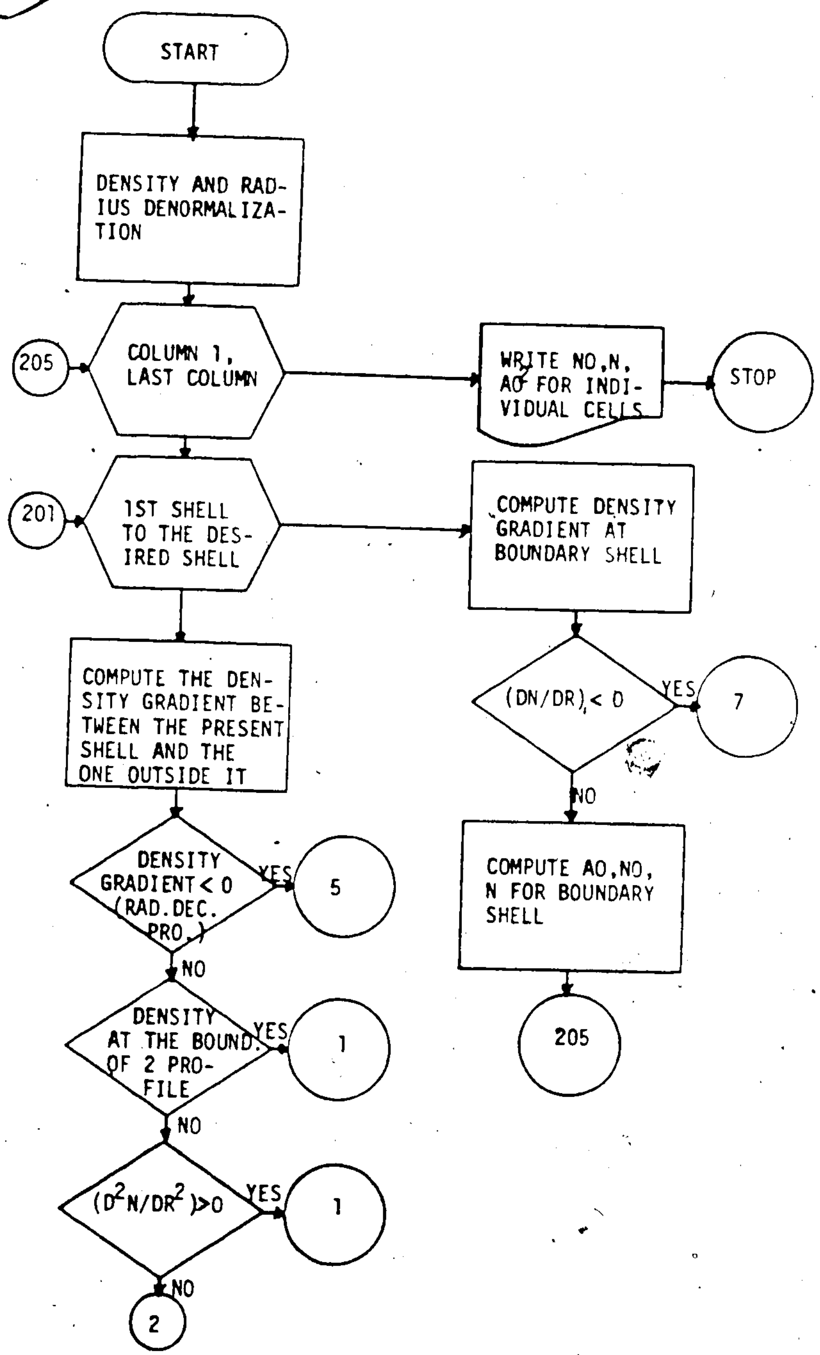


CHART 6. ALGORITHM FOR COMPUTING DENSITY GRADIENT

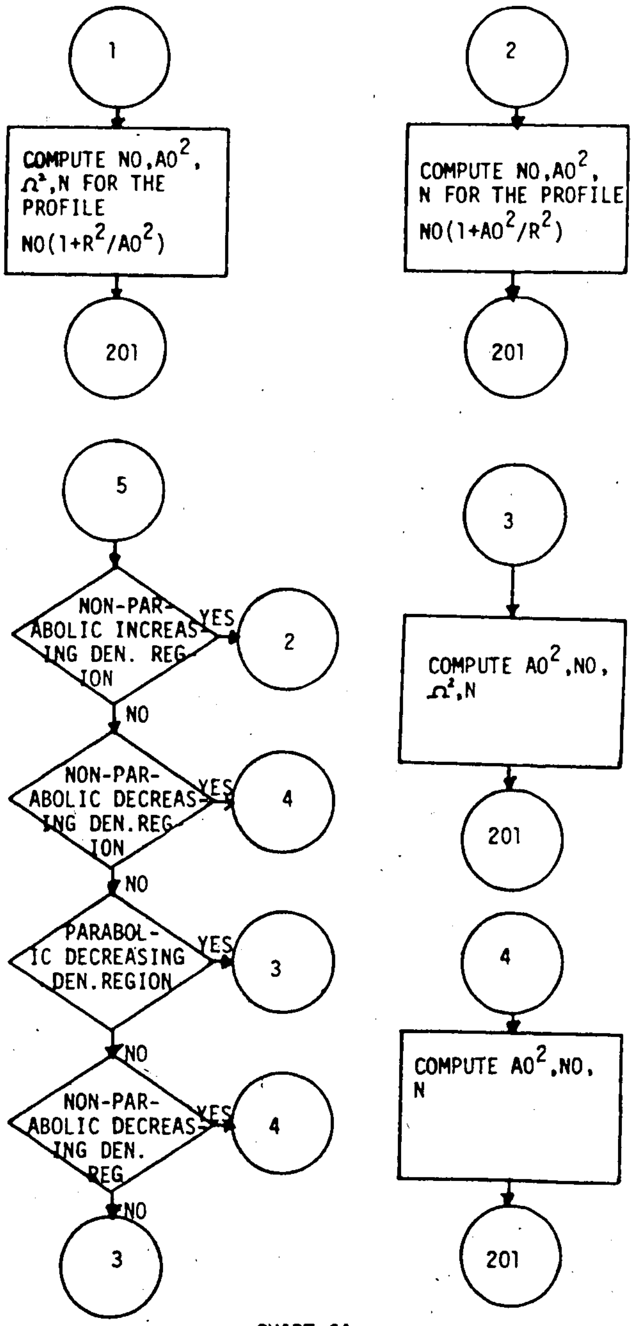

1
COMPUTE NO,AO^2, Ω^2,N FOR THE PROFILE NO(1+R^2/AO^2)
201
2
COMPUTE NO,AO^2, N FOR THE PROFILE NO(1+AO^2/R^2)
201
5
NON-PARABOLIC INCREASING DEN. REGION
YES
2
NO
NON-PARABOLIC DECREASING DEN.REGION
YES
4
NO
PARABOLIC DECREASING DEN.REGION
YES
3
NO
NON-PARABOLIC DECREASING DEN. REG
YES
4
NO
3
3
COMPUTE AO^2,NO, Ω^2,N
201
4
COMPUTE AO^2,NO, N
201

CHART 6A

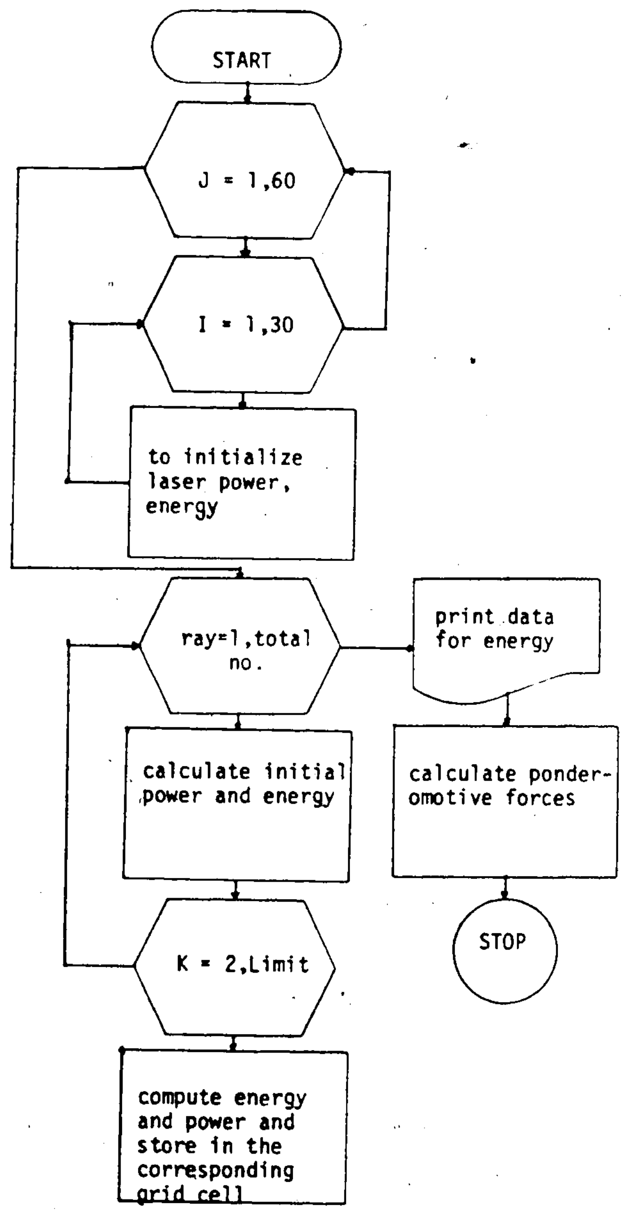


CHART 7. ENERGY ABSORPTION ALGORITHM

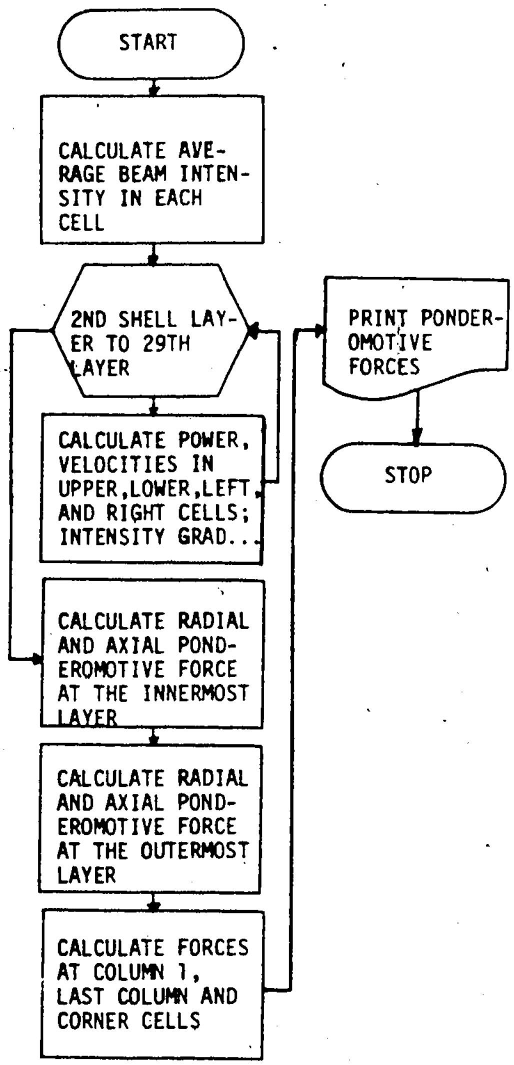


CHART 8. ALGORITHM FOR PONDEROMOTIVE FORCE

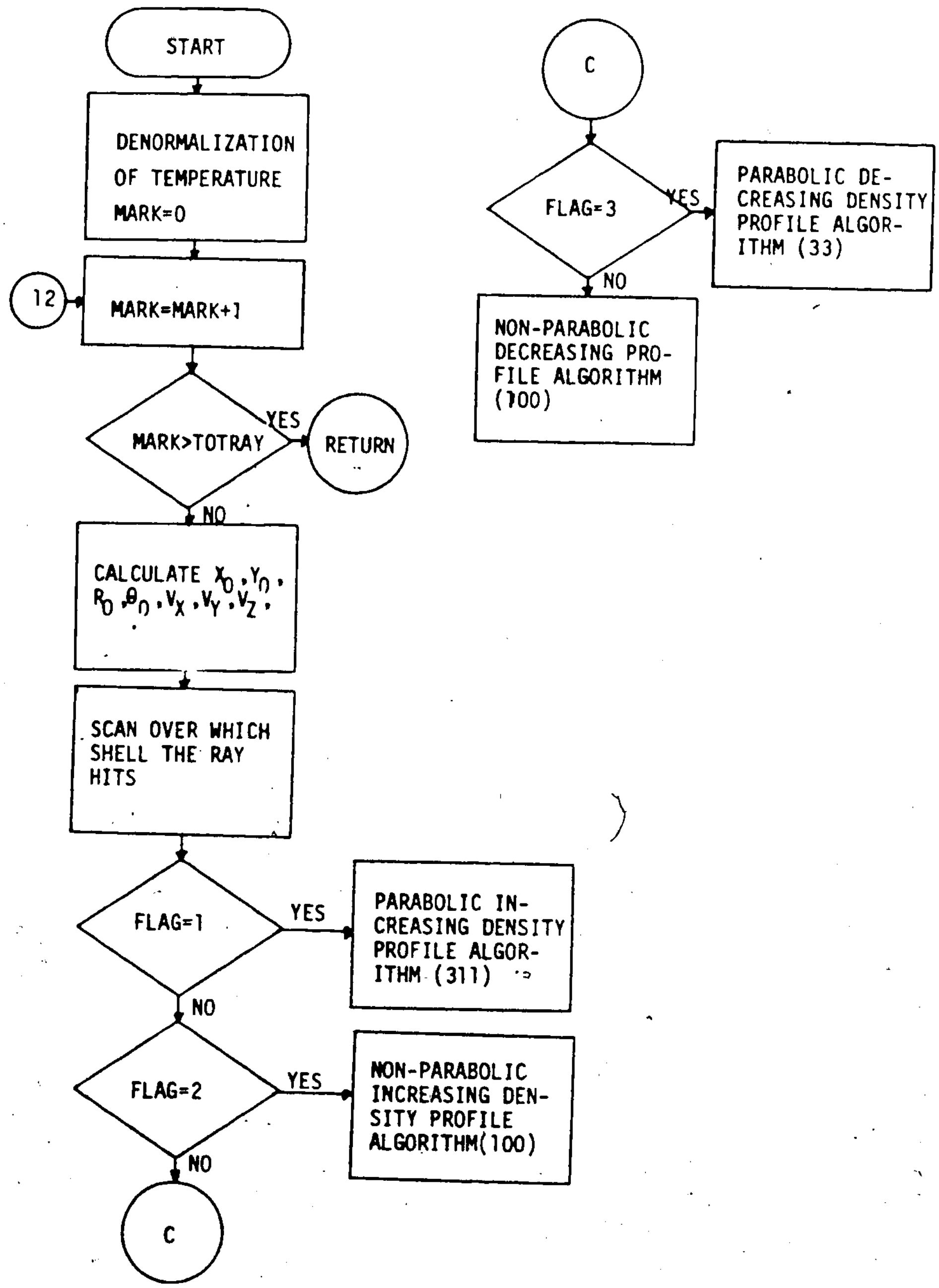


CHART 9. RAY TRACING PROGRAM ROUTINE

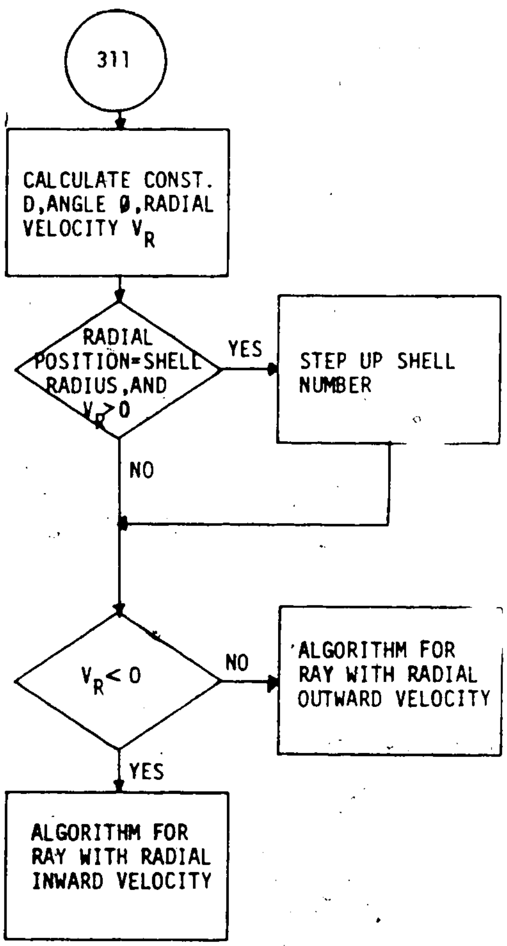


CHART 10. ALGORITHM FOR LOCATING RAY IN THE REGION WITH DENSITY $N_0(1+R^2/A0^2)$

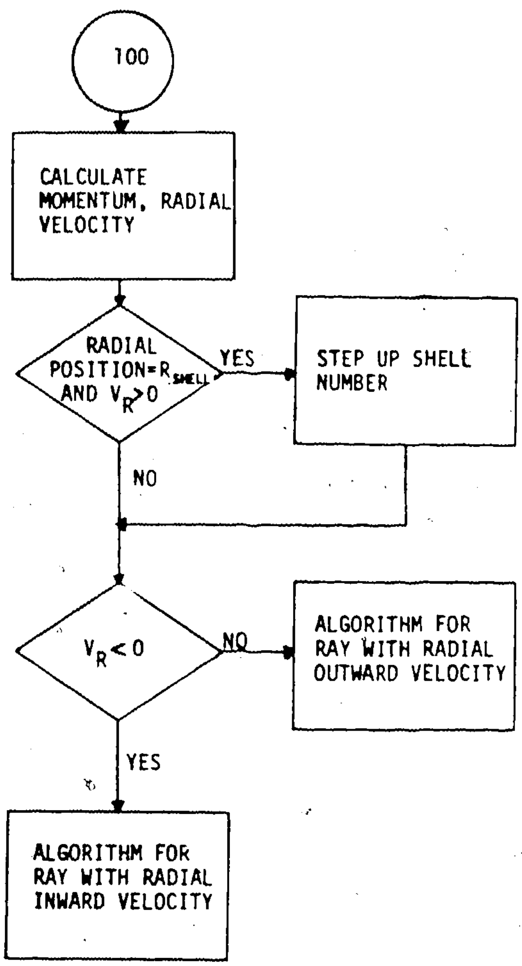


CHART 11. ALGORITHM FOR LOCATING RAY IN THE REGION WITH DENSITY $N_0(1+A0^2/R^2)$

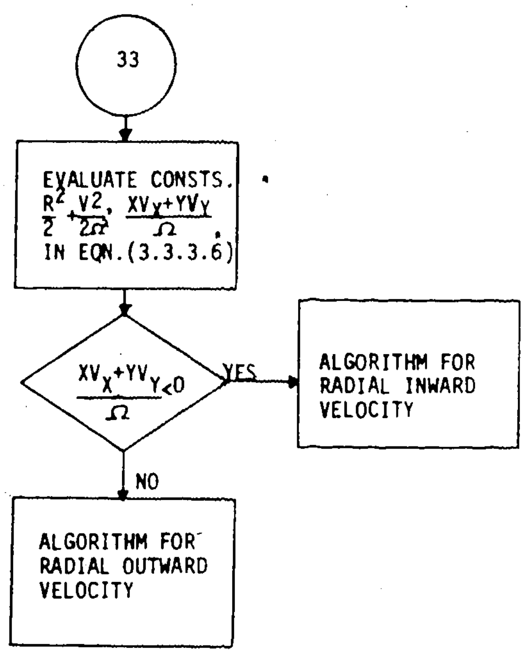


CHART 12. ALGORITHM LOCATING RAYS IN THE REGION WITH DENSITY $N_0(1-R^2/AO^2)$

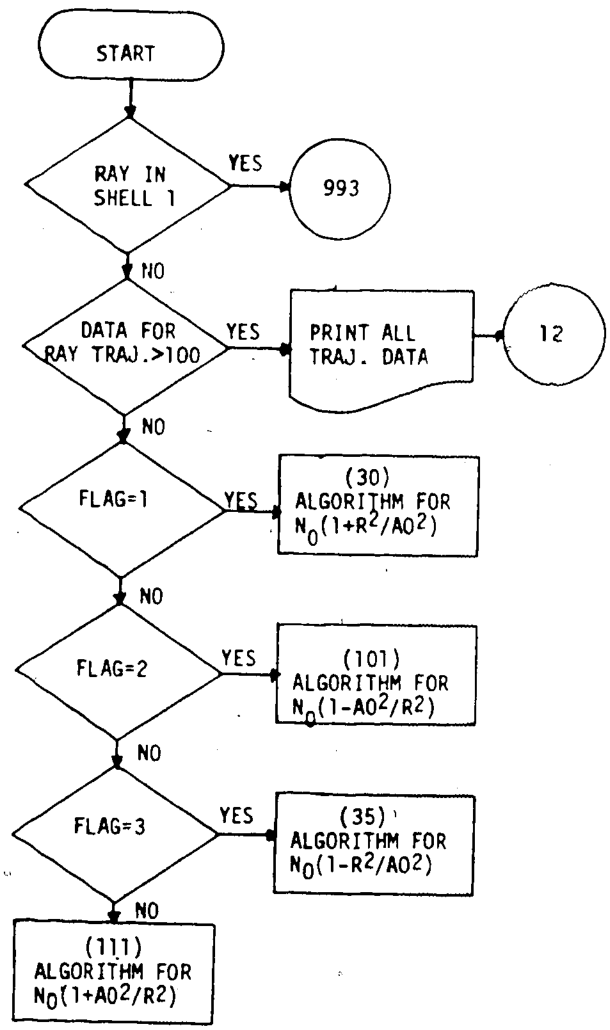


CHART 13. ALGORITHM FOR RAYS GOING RADIALLY OUTWARD

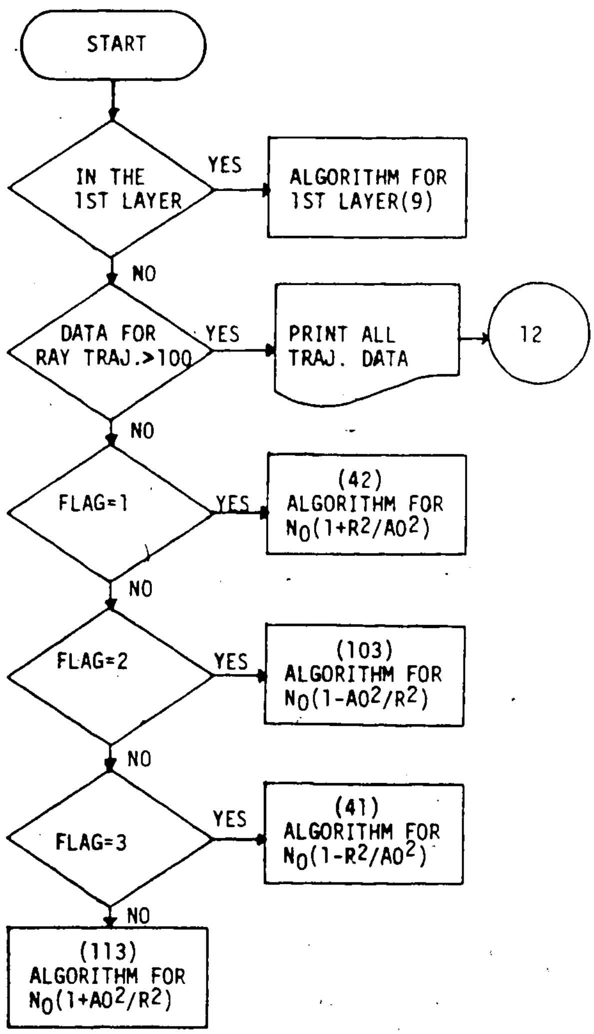


CHART 14. ALGORITHM FOR RAYS GOING RADIALLY OUTWARD

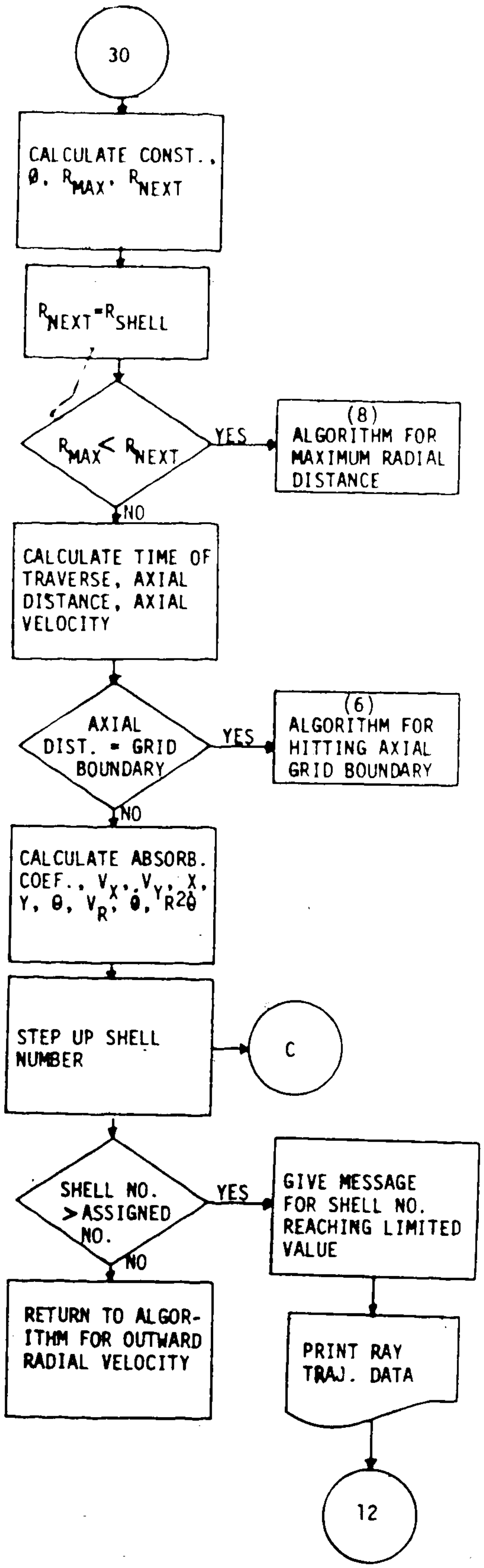


CHART 15. ALGORITHM FOR RAYS WITH RADIAL OUTWARD VELOCITY IN THE REGION WITH DENSITY $N_0(1+R^2/A0^2)$

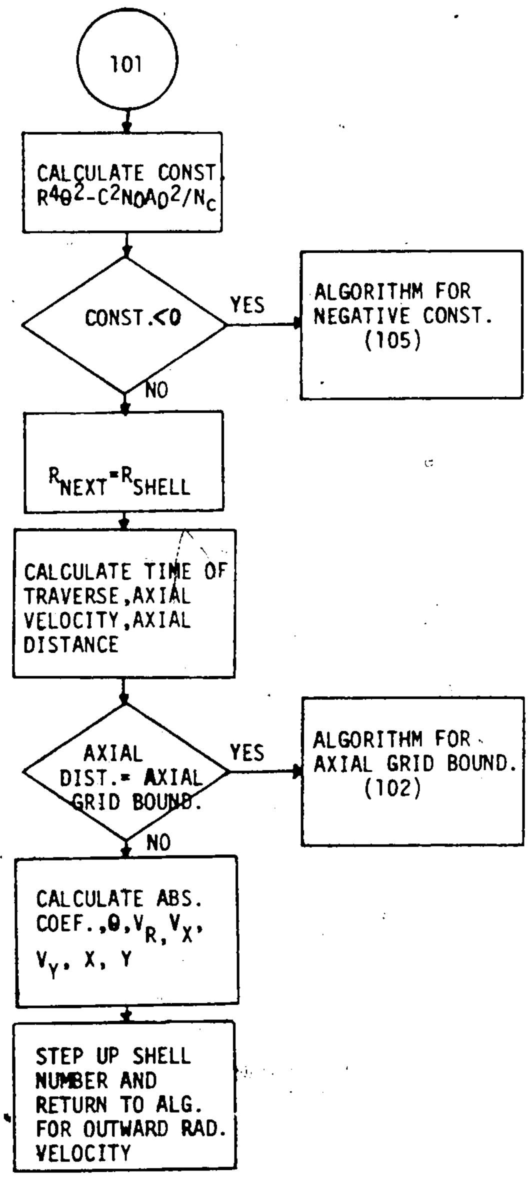


CHART 16. ALGORITHM FOR RAYS WITH RADIALLY OUTWARD VELOCITY IN THE REGION WITH DENSITY PROFILE $N_0(1+A0^2/R^2)$

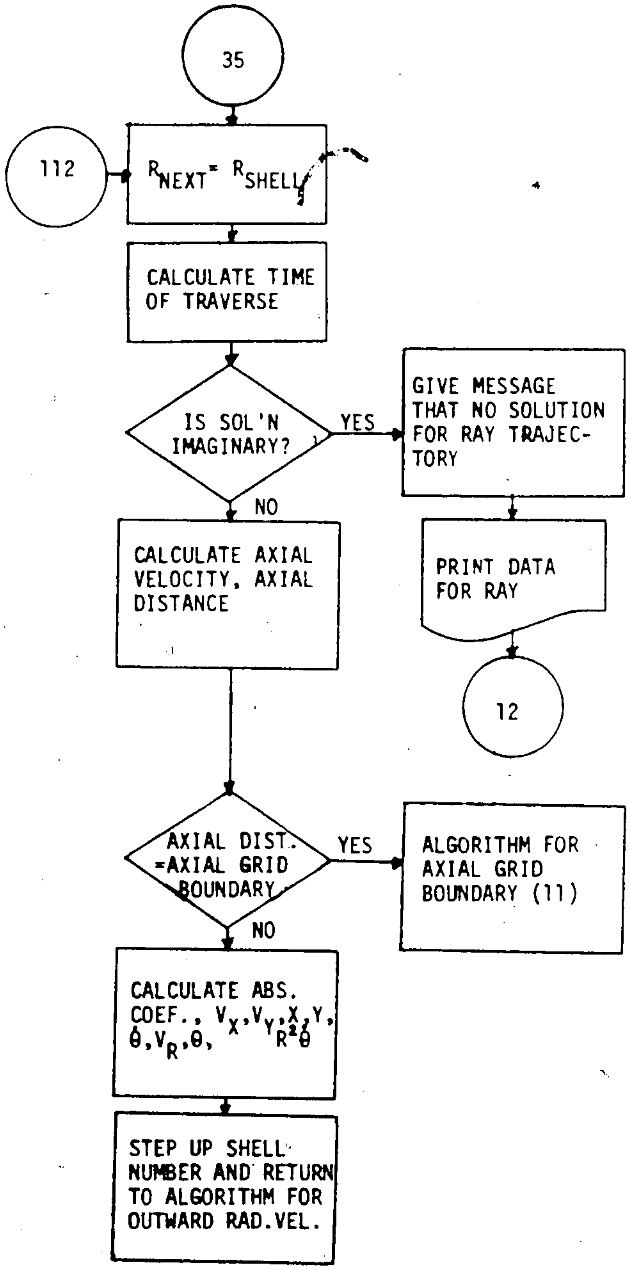


CHART 17. ALGORITHM FOR RAYS WITH RADIALLY OUTWARD VELOCITY IN THE REGION WITH DENSITY PROFILE $N_0(1-R^2/AO^2)$

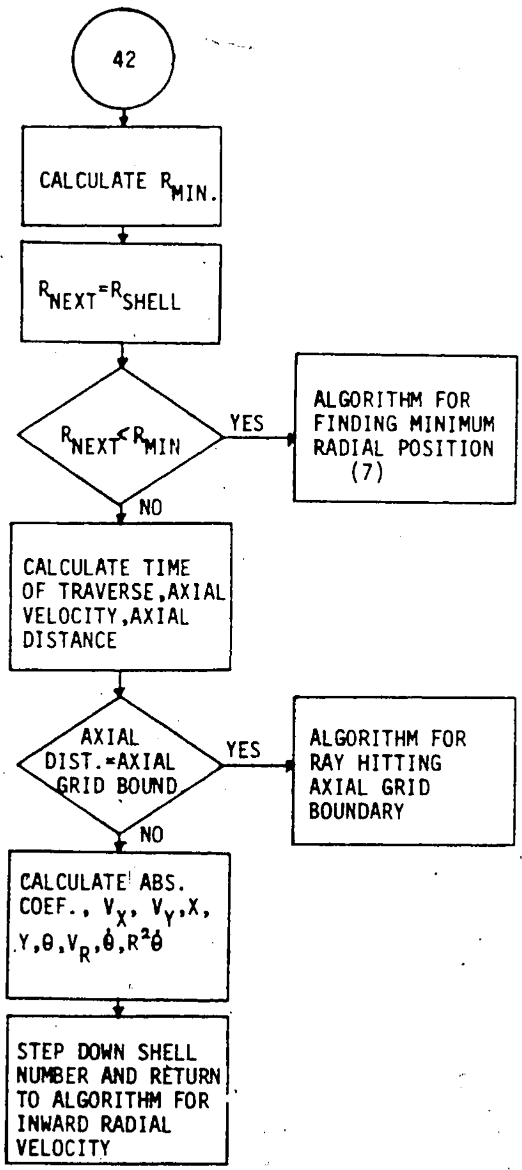


CHART 18. ALGORITHM FOR RAYS WITH RADIALLY OUTWARD VELOCITY IN THE REGION WITH DENSITY PROFILE $N_0(1+R^2/AO^2)$

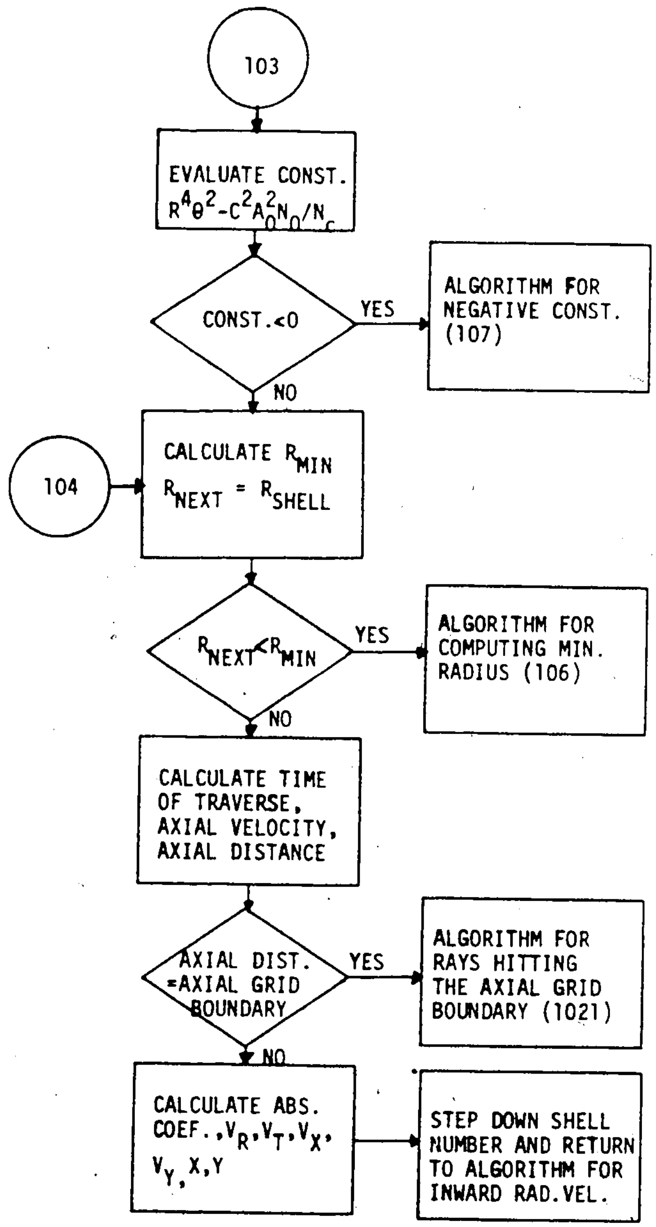


CHART 19. ALGORITHM FOR RAYS WITH RADIALLY INWARD VELOCITY IN THE REGION WITH DENSITY PROFILE $N_0(1-AO^2/R^2)$

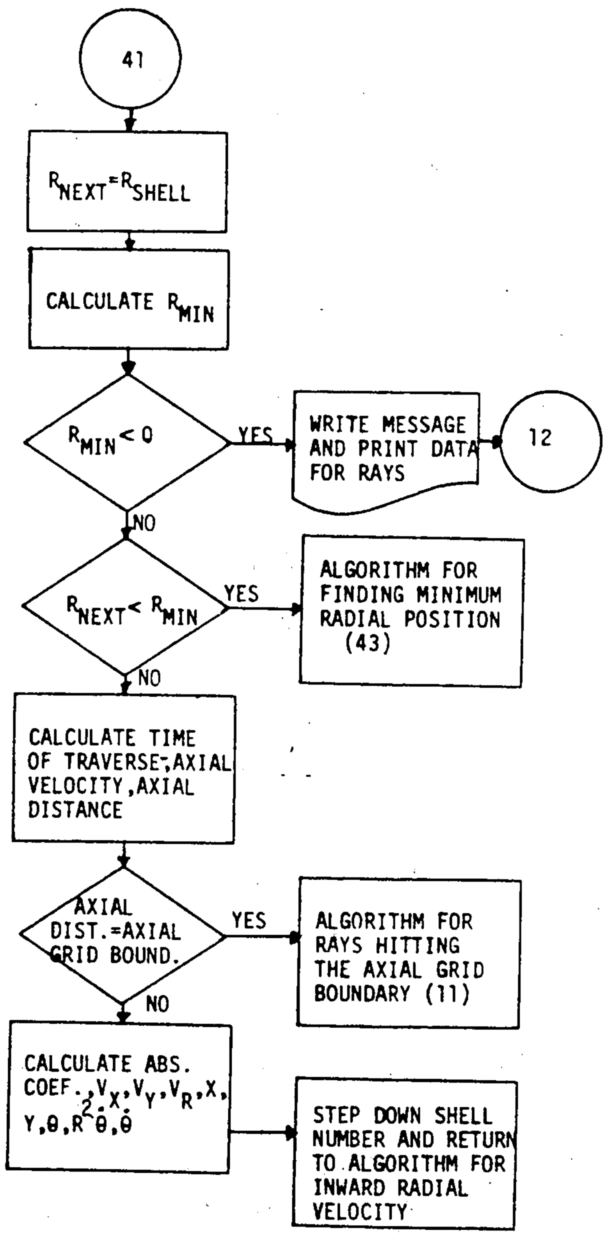


CHART 20. ALGORITHM FOR RAYS WITH RADIALLY INWARD VELOCITY IN THE REGION WITH DENSITY PROFILE $N_o(1-R^2/AO^2)$

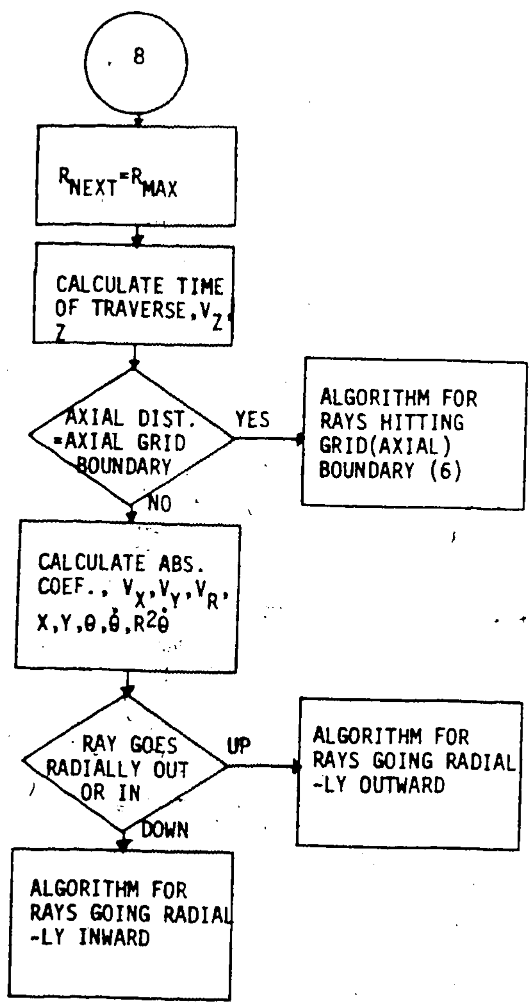


CHART 21. ALGORITHM FOR FINDING MAXIMUM RADIAL POSITION IN THE REGION WITH DENSITY PROFILE $N_0(1+R^2/A0^2)$

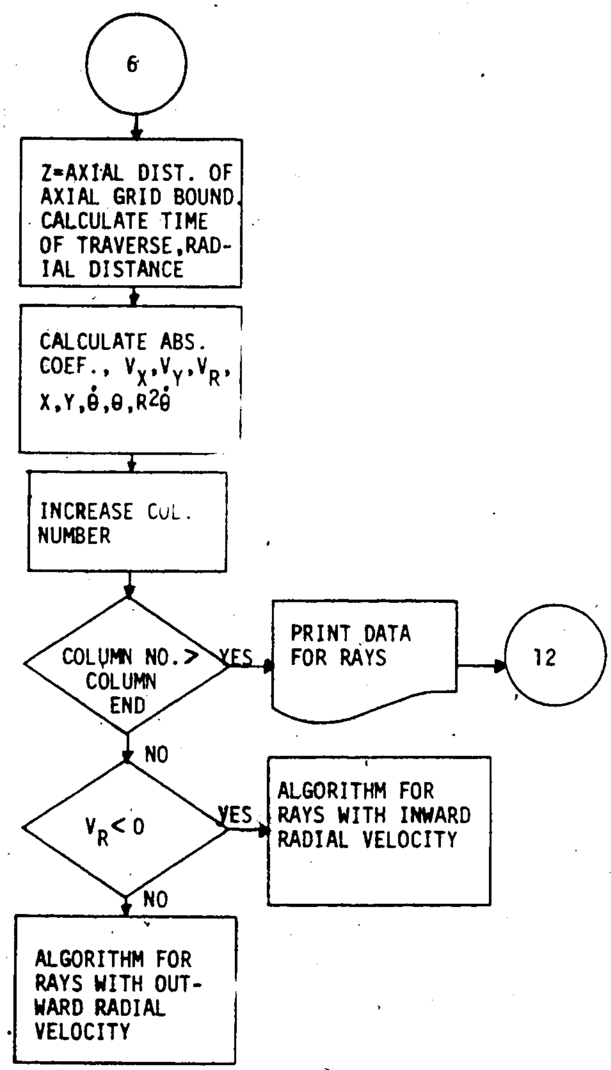


CHART 22. ALGORITHM FOR RAYS HITTING AT AXIAL GRID BOUNDARY

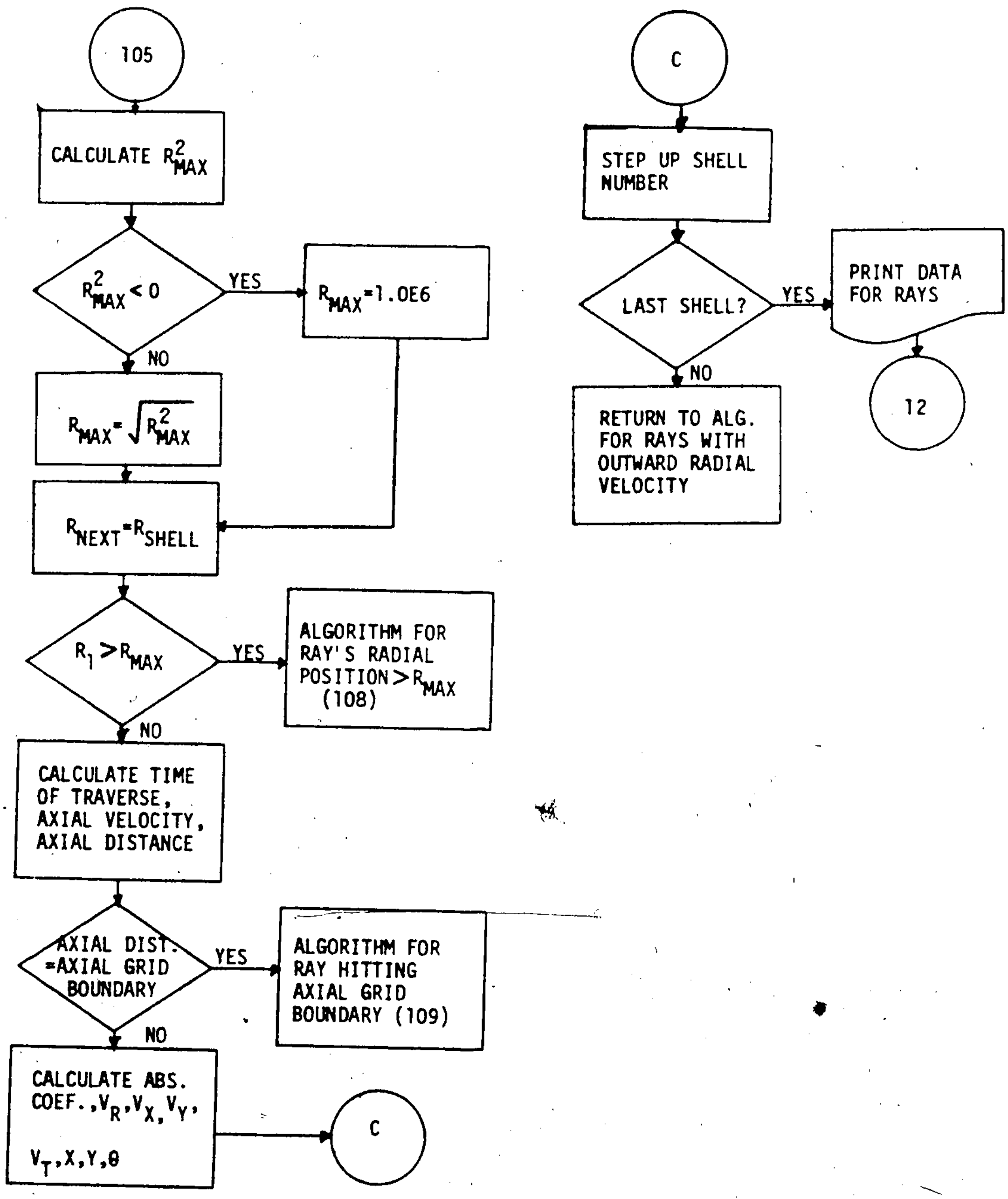


CHART 23. ALGORITHM FOR RAYS PROPAGATING IN THE REGION WITH DENSITY PROFILE NO(1-AO^2/R^2) [CONSTANT < 0, $V_R > 0$]

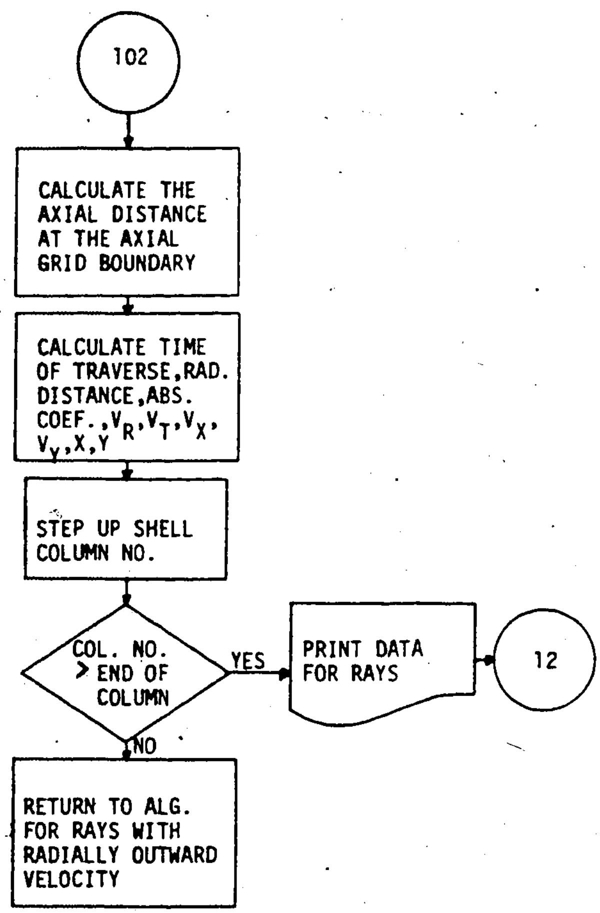


CHART 24. ALGORITHM FOR RAY HITTING AXIAL GRID BOUNDARY IN THE REGION WITH DENSITY PROFILE $NO(1-AO^2/R^2)$ [CONST > 0, $V_R > 0$]

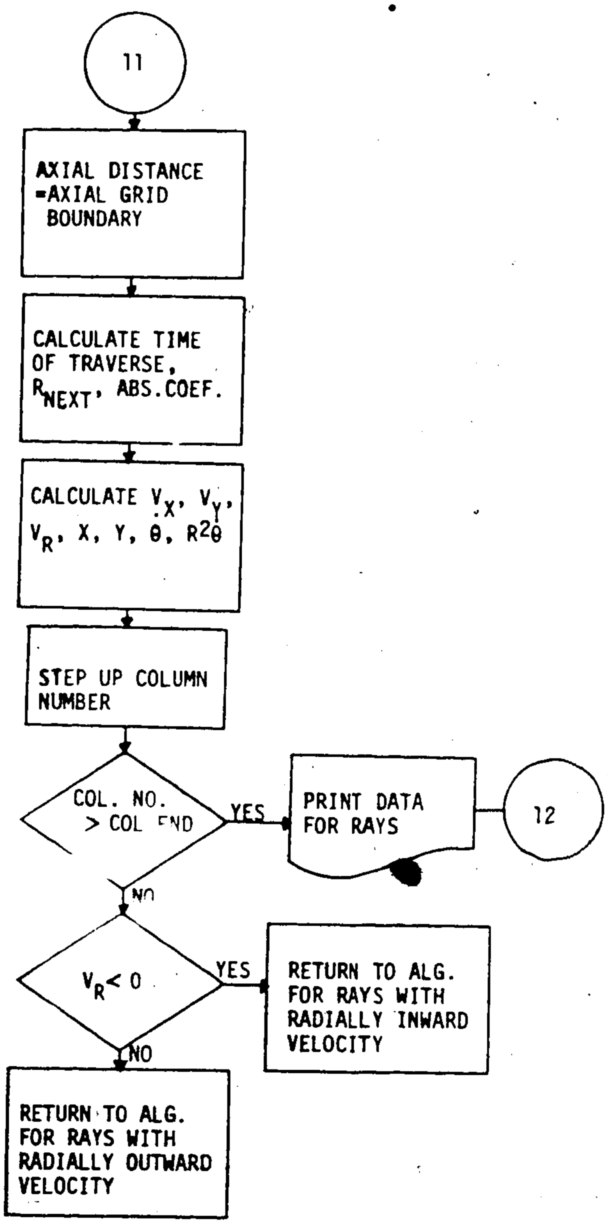


CHART 25. ALGORITHM FOR RAY HITTING AXIAL GRID BOUNDARY IN THE REGION WITH DENSITY PROFILE NO(1-R²/AO²)

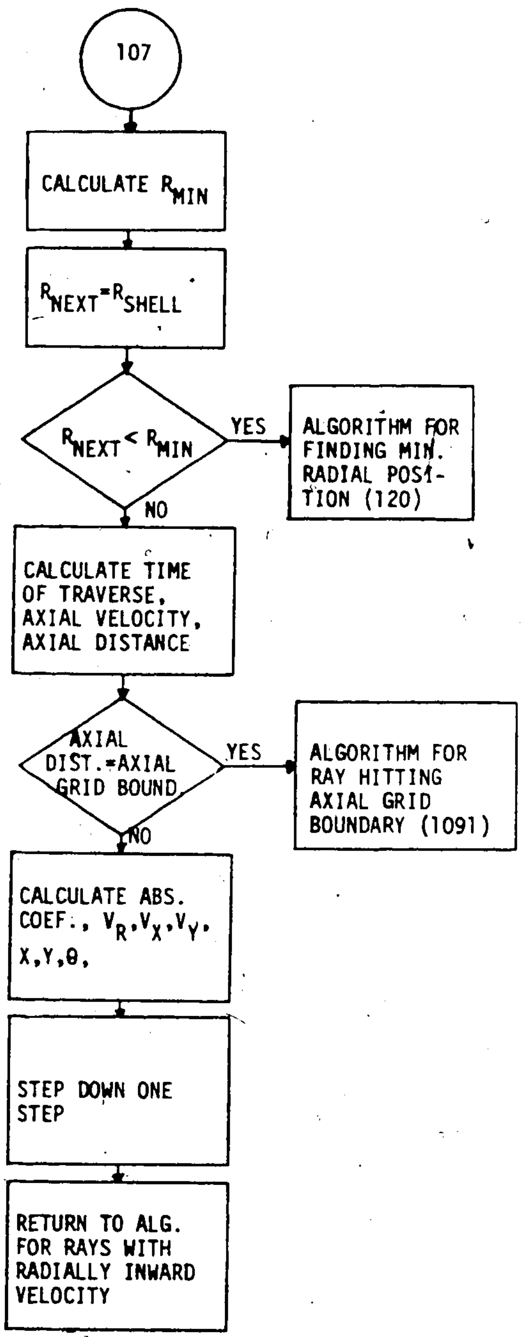


CHART 26. ALGORITHM FOR RAYS REACHING MINIMUM RADIAL DISTANCE IN THE REGION WITH DENSITY PROFILE $NO(1-AO^2/R^2)$ [CONSTANT <0]

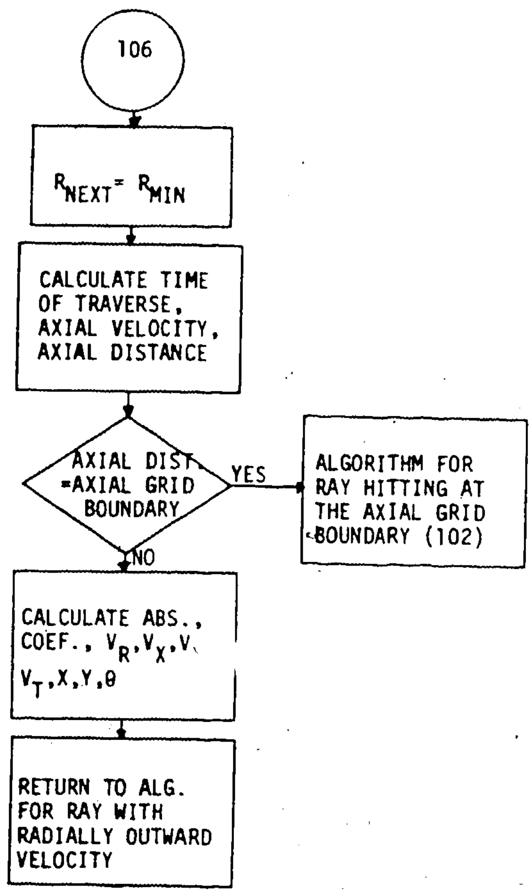


CHART 27. ALGORITHM FOR FINDING MINIMUM RADIAL DISTANCE IN THE REGION WITH DENSITY PROFILE $NO(1-AO^2/R^2)$ [CONSTANT > 0, $V_R < 0$]

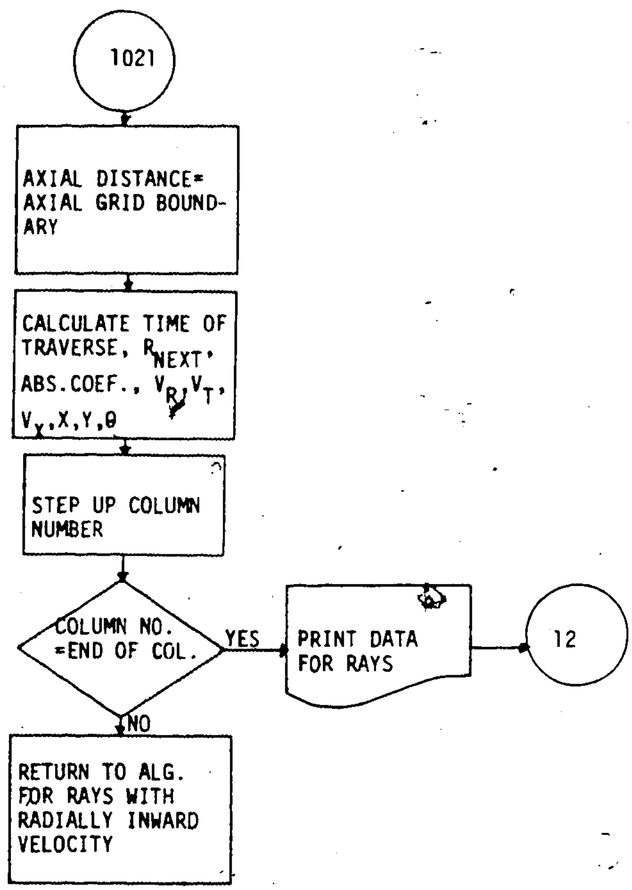


CHART 28. ALGORITHM FOR RAYS HITTING AXIAL GRID BOUNDARY IN THE REGION WITH DENSITY PROFILE $NO(1-AO^2/R^2)$ [CONSTANT$>$0, $V_R<0$]

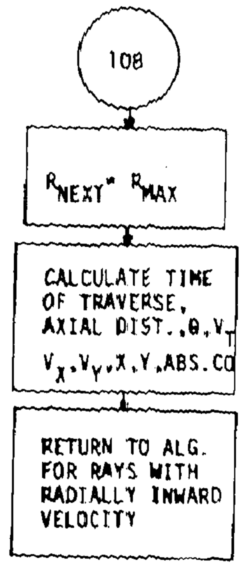


CHART 29. ALGORITHM FOR RAYS ATTAINING THE MAXIMUM RADIAL DISTANCE IN THE REGION WITH DENSITY PROFILE $NO(1-AO^2/R^2)$ [CONSTANT < 0, $V_R > 0$]

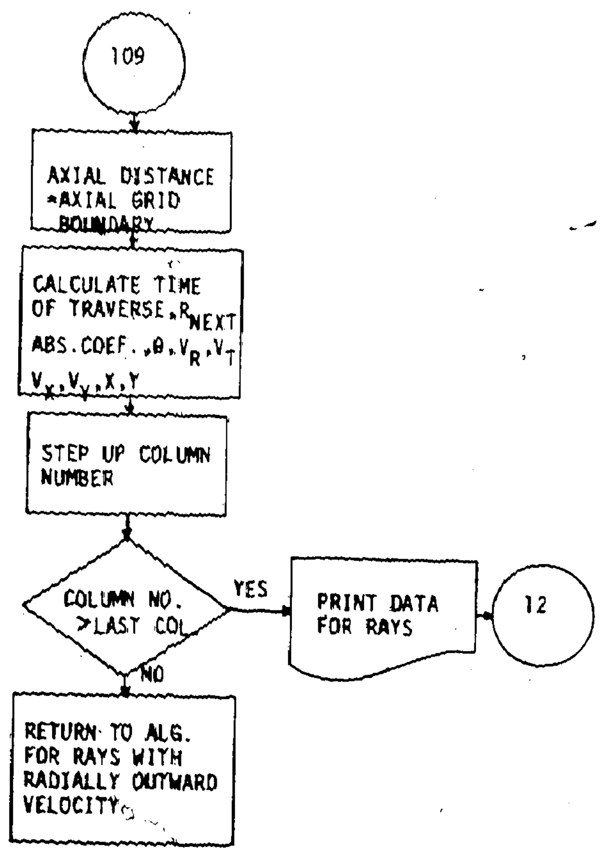


CHART 30. ALGORITHM FOR RAYS HITTING AT THE AXIAL GRID BOUNDARY IN THE REGION WITH DENSITY PROFILE $NO(1-AO^2/R^2)$ [CONSTANT < 0, $V_R > 0$]

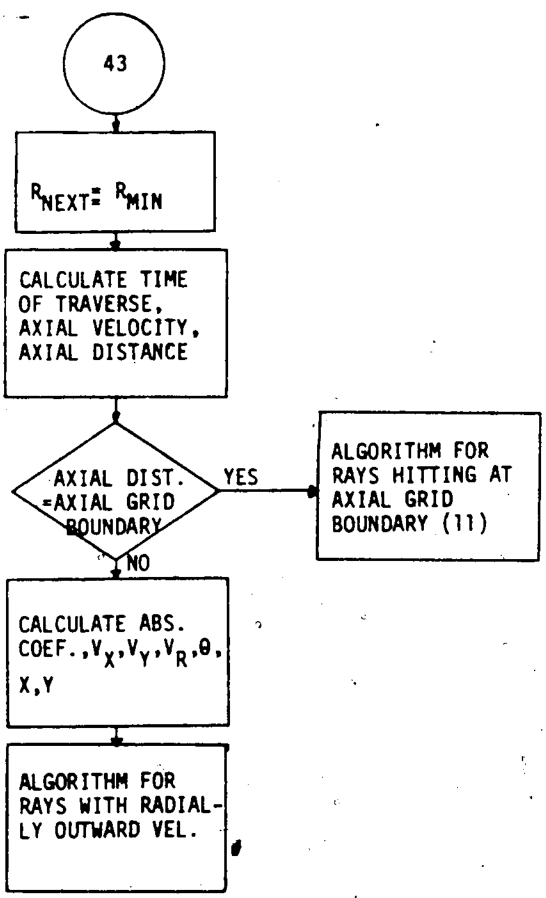


CHART 31. ALGORITHM FOR RAY ATTAINING MINIMUM RADIAL DISTANCE IN THE REGION WITH DENSITY PROFILE $NO(1-R^2/AO^2)$

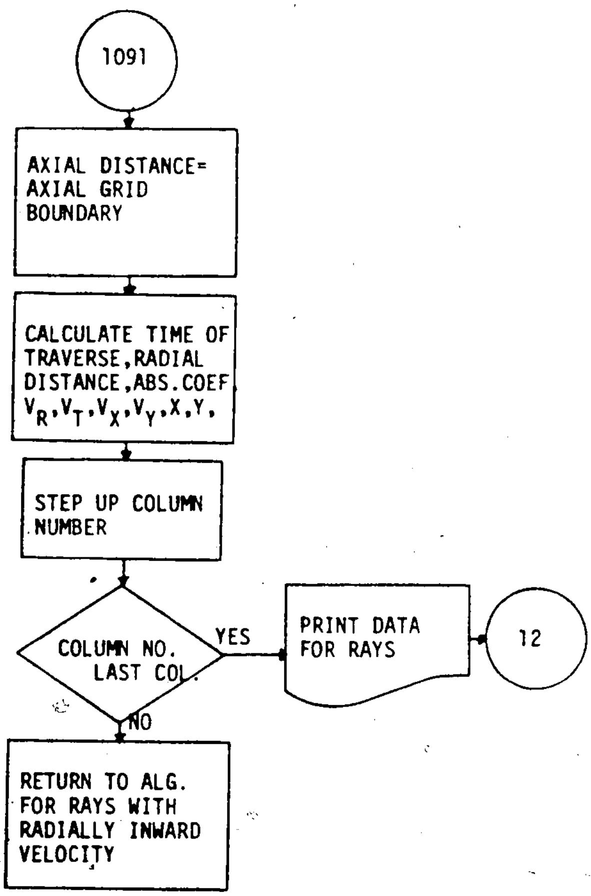


CHART 32. ALGORITHM FOR RAYS HITTING THE AXIAL GRID BOUNDARY IN THE REGION WITH DENSITY PROFILE $NO(1-AO^2/R^2)$ [CONSTANT < 0, V_R < 0]

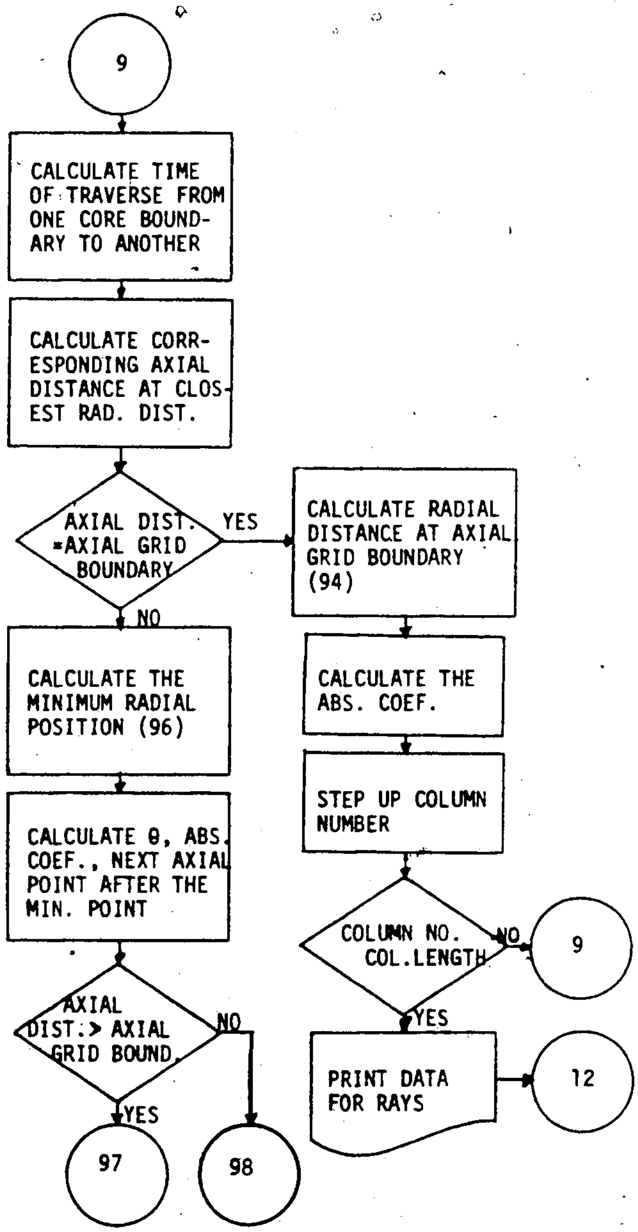


CHART 33. ALGORITHM FOR DETERMINING RAY LOCATIONS WHEN THE RAYS REACH THE INNERMOST CORE

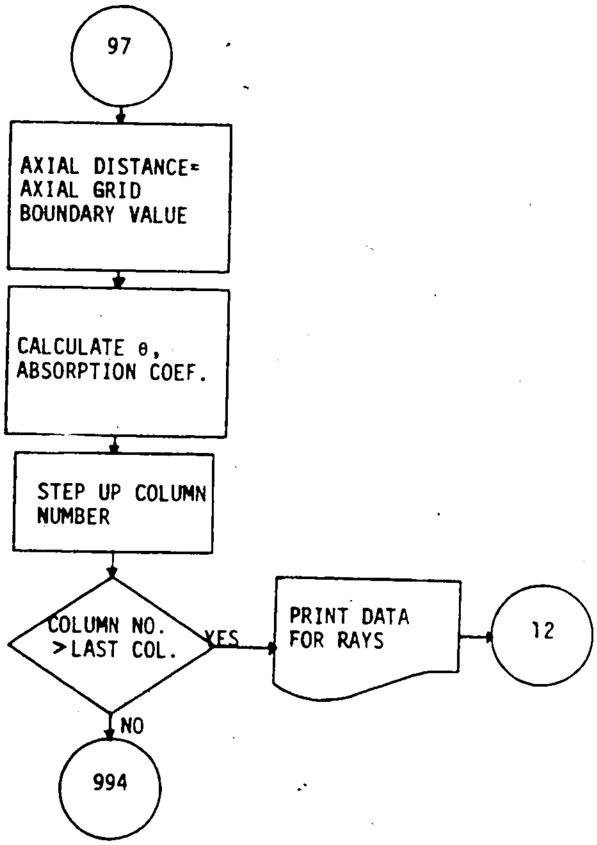


CHART 33A

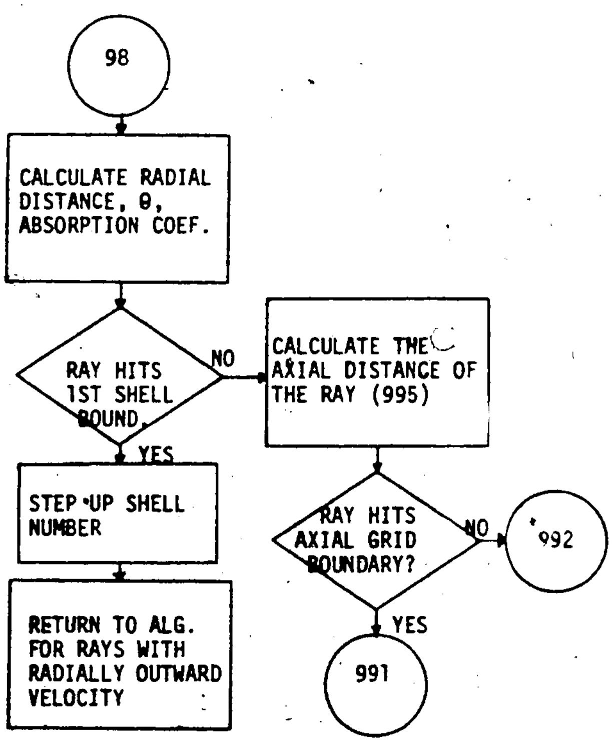


CHART 33B

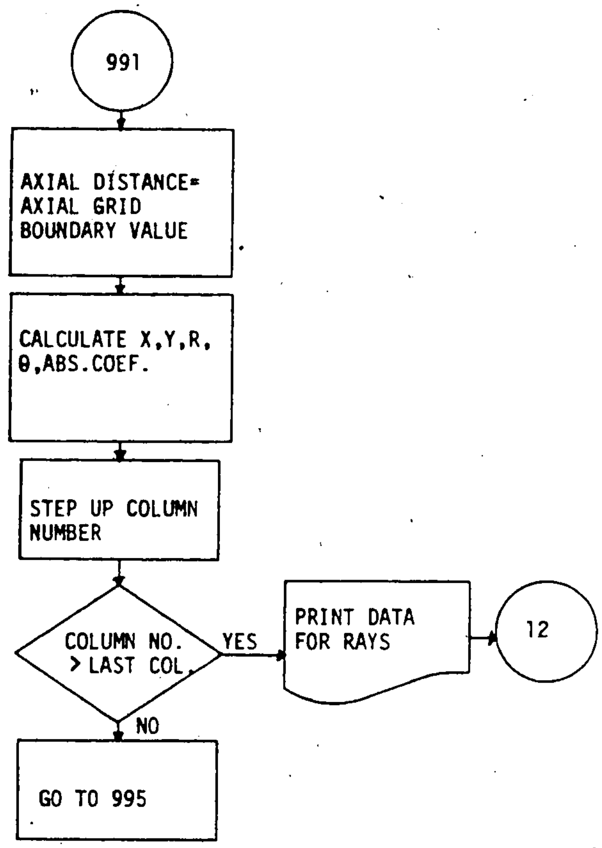


CHART 33C

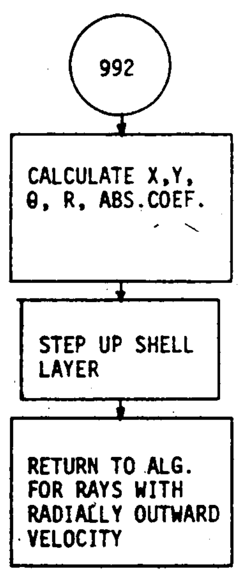


CHART 33D

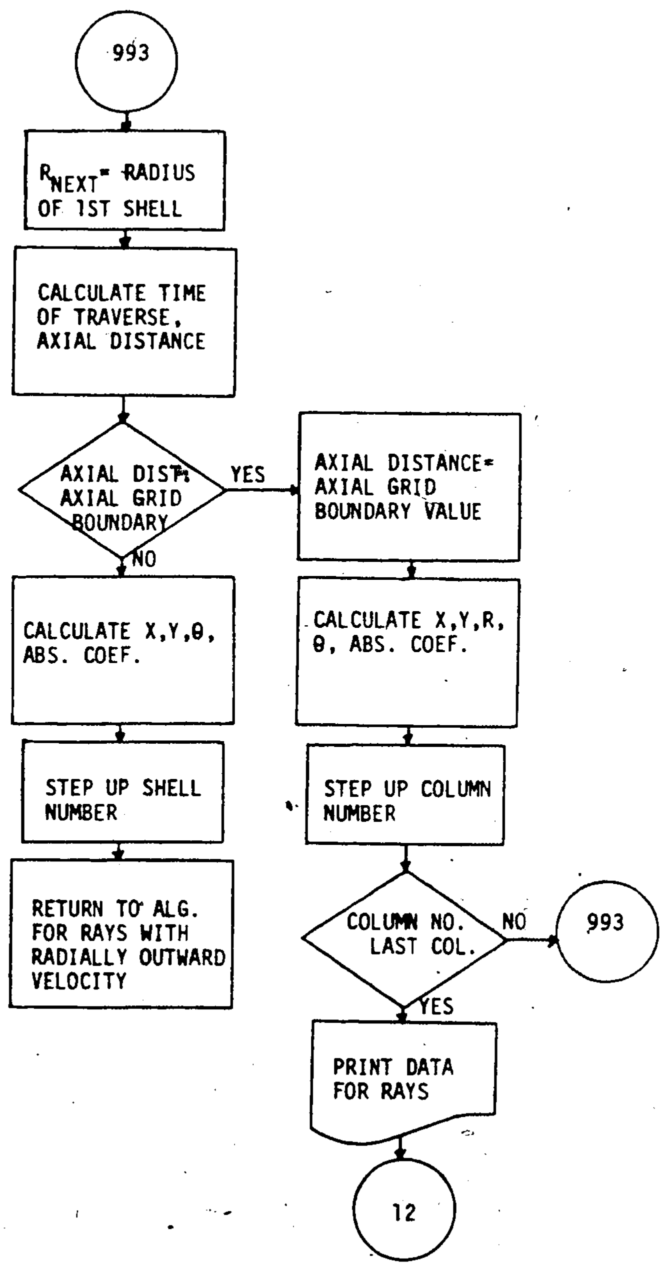


CHART 34. ALGORITHM FOR COMPUTING RAY TRAJECTORY WHEN RAY LOCATES INITIALLY IN THE INNER CORE

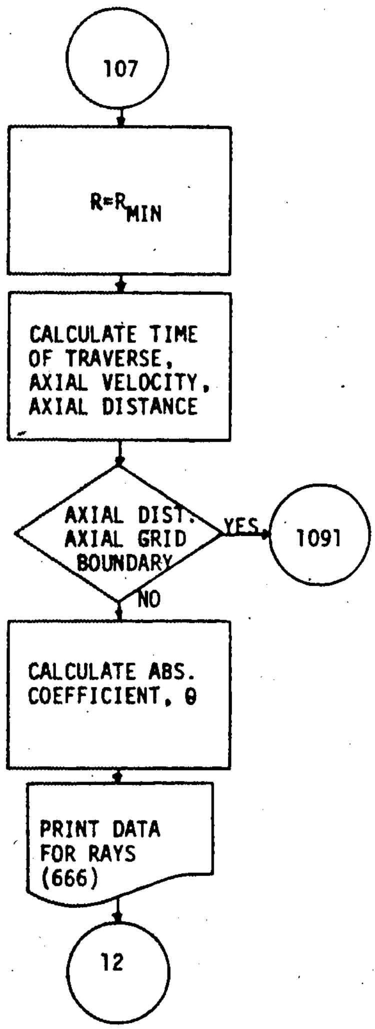


CHART 35. ALGORITHM FOR RAY REACHING MINIMUM RADIAL DISTANCE IN THE REGION WITH DENSITY PROFILE $NO(1-AO^2/R^2)$ [CONSTANT <0, $V_R<0$]

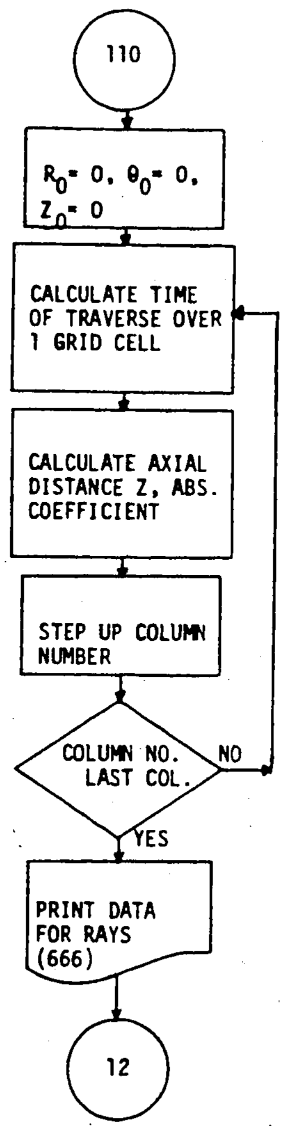


CHART 36. ALGORITHM FOR RAYS TRAVELLING ALONG THE AXIS OF PROPAGATION

```
C -------------------------------------------------------------------
C TESTING PROGRAM FOR THE RELATED SUBROUTINES
C -------------------------------------------------------------------
      NAMELIST /MAIDAT/TIME,STEP,NBOUND,PWR1,PWR2,PWR3,T1,T2,T3,T4,
     *NX,M,Z,RO,TO,LO,NO,VLAS,XMIN,LAMDA,DTL,MRMAX,XLAS,DX,TL,F,AO,
     *TOTRAY,EN1
      REAL LAMDA,LO,NO,R(30,60),N(30,60),TE(30,60),EP(30,60),PWR(30,60)
      INTEGER STEP,XMIN,Z,TOTRAY
      COMMON /LASP/L3,RLO,BMS,NCRIT,TL,DTL,MRMAX
      DATA PWR/1800*0.0/,EP/1800*0.0/
      READ(5,MAIDAT)
      WRITE(6,MAIDAT)
      CALL PWINIT(PWR1,PWR2,PWR3,T1,T2,T3,T4,TIME,EN1)
      CALL DRAY(F,LAMDA,AO,TOTRAY)
      IF (STEP.EQ.1) GOTO 1
      CALL MHDDEN(R,N,TE,NO,RO,NX,M)
      CALL GRID(R,NX,RO)
      CALL DNGR(R,N,LAMDA,NX,RO,NO)
      CALL RAYABS(M,NX,TE,TO,LAMDA,LO,Z,XLAS,VLAS,TIME,DX,XMIN)
    1 CALL ENERGY(TIME,STEP,XMIN,NX,XLAS,VLAS,DX,EP,PWR,N)
      STOP
      END
```

```
C -------------------------------------------------------------------
C PROGRAM FOR INITIALIZING THE INCIDENT BEAM POWER
C -------------------------------------------------------------------
      SUBROUTINE PWINIT(PWR1,PWR2,PWR3,T1,T2,T3,T4,TIME,EN1)
      REAL INDEX
      COMMON /DRAYP/XO(100),YO(100),THETAX(100),THETAY(100),P(100),TOTRA
     *Y,ENIN,P1,EN(100)
C
C TEMPORAL BEAM PROFILE IS SECTIONED INTO FOUR PARTS--T1,T2,T3,T4
C
      IF (TIME.GT.T1) GOTO 15
      P1=PWR1*TIME/T1
      ENO=P1*0.5*TIME
      ENIN=ENO-EN1
      EN1=ENO
      GOTO 1
   15 IF (TIME.GT.T2) GOTO 16
      P1=PWR1+(PWR2-PWR1)*(TIME-T1)/(T2-T1)
      ENO=(TIME+TIME-T1)*0.5*PWR1+((PWR2-PWR1)*(TIME-T1)/(T2-T1))*0.5*(T
     *IME-T1)
      ENIN=ENO-EN1
      EN1=ENO
      GOTO 1
   16 IF (TIME .GT. T3) GOTO 17
      P1=PWR2+(PWR3-PWR2)*(TIME-T2)/(T3-T2)
      ENO=(TIME+TIME-T1)*0.5*PWR1+(TIME-T1+TIME-T2)*(PWR2-PWR1)*0.5+0.5*
     *(TIME-T2)*(P1-PWR2)
      ENIN=ENO-EN1
      EN1=ENO
      GOTO 1
   17 IF (TIME.GT.T4) GOTO 18
      P1=PWR3-PWR3*(TIME-T3)/(T4-T3)
      ENO=(TIME+TIME-T1)*PWR1*0.5+(TIME-T1+TIME-T2)*0.5*(PWR2-PWR1)+(TIM
     *E-T2+TIME-T3)*0.5*(PWR3-PWR2)-(P1-PWR2)*0.5*(TIME-T3)
      ENIN=ENO-EN1
      EN1=ENO
      GOTO 1
   18 P1=0.0
    1 WRITE(6,601)P1,ENIN
  601 FORMAT(/'POWER=',E10.3,/,'INITIAL ENERGY=',E10.3)
      RETURN
      END
```

```
C-----------------------------------------------------------------------
C  SUBROUTINE FOR GENERATING A SET OF RAYS FOLLOWING THE GAUSSIAN
C  DISTRIBUTION
C
C  DEVICE  1:STORE R**2,Z
C          2:STORE AVERAGE R**2,Z
C          3:STORE NO. OF RAYS WITHIN REGION, RADIUS SQUARE
C          4:STORE X,Y CO-ORDINATE OF VARIOUS RAYS
C          6:LIST OF DATA FOR RAY POSITIONS,NO.OF RAYS IN DISTINCT REGION
C          7:STORE X,Y CO-ORDINATES OF VARIOUS RAYS AT LENS PLANE
C          8:STORE TRANSVERSE DIRECTIONS OF RAYS--THETAX,THETAY
C         10:STORE 3DB POWER POINTS,Z
C-----------------------------------------------------------------------
C
      SUBROUTINE DRAY(F,LAMDA,AO,TOTAL)
      DIMENSION X(100),Y(100),THETAX(100),THETAY(100),NPT(50),P(100),
     *RSQ(50),RNEW(50,100)
      DOUBLE PRECISION DSEED,DELT
      REAL LAMDA,AO
      INTEGER TOTRAY,TOTAL
      NAMELIST /RAYDAT/ N,DSEED,NRING,NPLANE,NEXTRA,NPLAN1,FACTOR
      COMMON /DRAYP/X,Y,THETAX,THETAY,P,TOTRAY,ENIN,P1,EN(100)
      READ(5,RAYDAT)
      TOTRAY=TOTAL
      WN=2*3.14159/(LAMDA*1.0E-4)*FACTOR
      WNAO=WN*AO
      FSPOT=F/WNAO
      DZ=F/(NPLANE*1.0-1.0)
      NRINP5=NRING+5
      WRITE(6,604)DSEED,FSPOT,AO
  604 FORMAT(/'DATA FROM PROGRAM DRAY',/'DSEED=',D15.8,/,'FOCAL SPOT SIZ
     *E=',E15.8,/,'BEAM RADIUS AT LENS PLANE=',E15.8)
C-----------------------------------------------------------------------
C         TO GENERATE ALL RAY LOCATIONS AND DIRECTIONS AT LENS PLANE
C         THE CHOICE OF RAY LOCATIONS AND DIRECTIONS OBEYS A GAUSSIA
C         N DISTRIBUTION. SUBROUTINE GGNML GIVES A RANGE OF RANDOM
C         NORMAL DEVIATES WITHIN THE RANGE (0,1)
C-----------------------------------------------------------------------
      CALL GGNML(DSEED,N,X)
      CALL GGNML(DSEED,N,Y)
      CALL GGNML(DSEED,N,THETAX)
      CALL GGNML(DSEED,N,THETAY)
C-----------------------------------------------------------------------
C         TO CONVERT THE NORMALIZED VALUES TO ACTUAL VALUES
C         RAY DIRECTIONS ALWAYS TAKE OPPOSITE SIGNS TO LOCATIONS
C-----------------------------------------------------------------------
      DO 201 I=1,N
      X(I)=X(I)*AO/1.4142136
      THETAX(I)=THETAX(I)/WNAO/1.4142136-X(I)/F
      Y(I)=Y(I)*AO/1.4142136
      THETAY(I)=THETAY(I)/WNAO/1.4142136-Y(I)/F
      P(I)=P1/(N*1.0)
      EN(I)=ENIN/(N*1.0)
      WRITE(4,401)X(I),Y(I)
      WRITE(8,401)THETAX(I),THETAY(I)
  401 FORMAT(2E18.10)
  201 CONTINUE
C-----------------------------------------------------------------------
C         TO TRACE THE RAY PATH (R**2) ALONG AXIS
C-----------------------------------------------------------------------
      NPLANE=NPLANE+NEXTRA
      FINALZ=(NPLANE*1.0-1.0)*DZ
      DO 202 J=1,N
      DO 203 I=NPLAN1,NPLANE
      DELT=(I-1)*DZ*1.0D0
      XNEW=X(J)+THETAX(J)*DELT
      YNEW=Y(J)+THETAY(J)*DELT
      RNEW(I-NPLAN1+1,J)=XNEW**2+YNEW**2
      DELT1=SNGL(DELT)
      IF (ABS(DELT1-FINALZ).LT.10E-3) WRITE(7,701)XNEW,YNEW
  701 FORMAT(2E18.10)
```

```
          WRITE(1,101)RNEW(I-NPLAN1+1,J),DELT
      101 FORMAT(2E18.10)
      203 CONTINUE
          X(J)=XNEW
          Y(J)=YNEW
      202 CONTINUE
C---------------------------------------------------------------------
C           TO CALCULATE THE AVERAGE SPOT SIZE AT EACH AXIAL PLANE
C           AND NO. OF RAY POINTS WITHIN A DEFINED BEAM AREA AT A
C           PARTICULAR PLANE
C---------------------------------------------------------------------
          DO 204 I=NPLAN1,NPLANE
          RSQSUM=0.0
          DELT=(I-1)*DZ*1.0D0
          WRITE(6,606)
      606 FORMAT(/' AVE. SPOT SIZE',5X,' AXIAL DISTANCE')
C---------------------------------------------------------------------
C           AVERAGE SPOTSIZE SQUARE
C---------------------------------------------------------------------
          DO 205 J=1,N
      205 RSQSUM=RSQSUM+RNEW(I-NPLAN1+1,J)
          RSQAVE=RSQSUM/(N*1.0)
[illegible](2,101)RSQAVE,DELT
[illegible](6,101) RSQAVE,DELT
[illegible]-----------------------------------------------------------
[illegible]RESPONDING RADIAL DISTANCE
[illegible]-----------------------------------------------------------
[illegible]=FSPOT**2+(DELT-F)**2/WN**2/FSPOT**2
[illegible]TSQ=SPOTSQ/NRING
[illegible] 206 K=1,NRINP5
[illegible]SQ(K)=K*SP[illegible]SQ
          NPT(K)=0
      206 CONTINUE
C---------------------------------------------------------------------
C           TO CATEGORIZE RAY LOCATIONS INTO VARIOUS REGIONS
C---------------------------------------------------------------------
          DO 207 L=1,N
          DO 208 M=1,NRINP5
          IF (RNEW(I-NPLAN1+1,[illegible].GT.RSQ(M)) GOTO 208
          NPT(M)=NPT(M)+1
          GOTO 207
      208 CONTINUE
      207 CONTINUE
C---------------------------------------------------------------------
C           SUM UP POINTS AT CORRESPONDING RADIUS
C---------------------------------------------------------------------
          WRITE(6,607)
      607 FORMAT(//'    RAY PTS. DIST.   ','   RADIUS SQUARE   ',
         *'THEORETICAL POINTS')
          SUM=0.0
          VAR=0.0
          DO 209 II=1,NRINP5
          SUM=SUM+NPT(II)
          RAD=N*(1-EXP(-RSQ(II)/RSQAVE))
          VAR=VAR+(RAD-SUM)**2/RAD**2
          WRITE(3,101)SUM,RSQ(II)
          IF (ABS(SUM-63.0).LE.2.0) WRITE(10,101)RSQ(II),DELT
          WRITE(6,605)SUM,RSQ(II),RAD
      605 FORMAT(3E18.10)
      209 CONTINUE
          VAR=VAR/NRINP5
          WRITE(6,608)VAR
      608 FORMAT(/'VARIANCE=',E15.8)
      204 CONTINUE
          RETURN
          END
```

```
C----------------------------------------------------------------------
C SUBROUTINE FOR CALCULATING DENSITIES AND TEMPERATURE IN EACH CELL
C AND THE CORRESPONDING RADIUS
C
C   DEVICE 2:DENSITY VALUE,NORMALIZED RADIUS,
C          6:LISTING OF DATA
C         11:TAPE DEVICE
C----------------------------------------------------------------------
C
      SUBROUTINE MHDDEN(R,N,TE,AXIDEN,PASRAD,ZSTEP,PLASEL)
      REAL R(30,60),N(30,60),TE(30,60)
      INTEGER ZSTEP,PLASEL
      DR=PASRAD/(PLASEL*1.0)
      A1=0.337769221**2
      A2=0.185919614**2
      A3=1.887104451**2
      A4=0.356427765**2
      DO 201 I=1,6
      R(I,1)=I*DR/PASRAD
C----------------------------------------------------------------------
C          CCR--CELL CENTRE RADIUS
C          N-- PARABOLIC DENSITY PROFILE
C----------------------------------------------------------------------
      CCR=(I-0.5)*DR
      N(I,1)=(1+CCR**2/A1)*0.997362083E18
      TE(I,1)=50/(1+EXP((CCR-2.0)/0.1))
  201 CONTINUE
      DO 204 I=7,9
      R(I,1)=I*DR/PASRAD
      CCR=(I-0.5)*DR
      N(I,1)=(1-A2/CCR**2)*3.059522465E18
      TE(I,1)=50/(1+EXP((CCR-2.0)/0.1))
  204 CONTINUE
      DO 205 I=10,14
      R(I,1)=I*DR/PASRAD
      CCR=(I-0.5)*DR
      N(I,1)=(1-CCR**2/A3)*2.588205127E18
      TE(I,1)=50/(1+EXP((CCR-2.0)/0.1))
  205 CONTINUE
      DO 206 I=15,30
      R(I,1)=I*DR/PASRAD
      CCR=(I-0.5)*DR
      N(I,1)=(1+A4/CCR**2)*1.763802596E18
      TE(I,1)=50/(1+EXP((CCR-2.0)/0.1))
  206 CONTINUE
C----------------------------------------------------------------------
C ASSIGN RADIUS AND DENSITY IN THE AXIAL DIRECTION
C----------------------------------------------------------------------
      DO 202 J=2,ZSTEP
      DO 203 I=1,PLASEL
      R(I,J)=R(I,1)
      N(I,J)=N(I,1)
      TE(I,J)=TE(I,1)
  203 CONTINUE
  202 CONTINUE
      WRITE(6,604)
  604 FORMAT(/'DATA FROM PROGRAM:RICKMHDDEN')
      WRITE(6,605)AXIDEN,PASRAD,ZSTEP,PLASEL
  605 FORMAT(/'AXIAL DENSITY(NORMALIZED)=',E15.8,/,'PLASMA RADIUS=',E15.
     *8,/,'AXIAL STEPS=',I3,/,'PLASMA SHELLS=',I3/)
      WRITE(6,602)
  602 FORMAT('  SHELL RADIUS(NORMALIZED) ',5X,' SHELL DENSITY ',5X,'SHEL
     *L TEMPERATURE')
      WRITE(6,603)(R(I,1),N(I,1),TE(I,1),I=1,PLASEL)
  603 FORMAT(E15.8,15X,E15.8,7X,E15.8)
      DO 20 I=1,PLASEL
      RN=R(I,1)*PASRAD-0.5*DR
      WRITE(2,103)N(I,1),RN
  103 FORMAT(2E18.10)
   20 CONTINUE
```

```
C
C TO STORE UNFORMATED DATA ON TAPE
C
      WRITE(11) N
      RETURN
      END
```

```
C ----------------------------------------------------------------------
C ALGORITHM FOR DIVIDING THE PLASMA LAYERS INTO FINER
C SHELLS (2 SUBSHELLS FOR EACH LAYER)
C ----------------------------------------------------------------------
      SUBROUTINE GRID(R,NX,RO)
      DIMENSION RDENOR(30,60),R(30,60)
      REAL N
      NAMELIST /GRIDAT/LASHEL
C ----------------------------------------------------------------------
C           LASHEL---NO. OF PLASMA SHELLS TAKEN FOR A FINER DIVISION   .
C ----------------------------------------------------------------------
      COMMON /GRIDP/R2(60,60),OMEGA(60,60),LASHEL
      READ(5,GRIDAT)
      WRITE(6,GRIDAT)
      DO 208 J=1,NX
      DO 209 I=1,LASHEL
      RDENOR(I,J)=R(I,J)*RO
  209 CONTINUE
  208 CONTINUE
      DO 201 J=1,NX
C ----------------------------------------------------------------------
C           DIVIDE EACH OF THE 3RD AND ABOVE PLASMA LAYER INTO 2 SMALLER
C           LAYERS
C ----------------------------------------------------------------------
      DO 202 I=3,LASHEL
      DRI=RDENOR(I,J)-RDENOR(I-1,J)
      DRIM1=RDENOR(I-1,J)-RDENOR(I-2,J)
      DRS=DRI+DRIM1
      RMID=(DRI/DRS)*RDENOR(I-1,J)+(DRIM1/DRS)*RDENOR(I,J)
      I2M1=2*I-1
      I2M2=2*I-2
      R2(I2M1,J)=RMID
      R2(I2M2,J)=RDENOR(I-1,J)
  202 CONTINUE
C ----------------------------------------------------------------------
C           DIVIDE THE 1ST LAYER INTO 2 SMALLER LAYERS
C ----------------------------------------------------------------------
      R2(1,J)=0.5*RDENOR(1,J)
C ----------------------------------------------------------------------
C           DIVIDE THE 2ND LAYER INTO 2 SMALLER LAYERS
C ----------------------------------------------------------------------
      R2(2,J)=RDENOR(1,J)
      R2(3,J)=0.5*(RDENOR(2,J)+RDENOR(1,J))
      R2(60,J)=RDENOR(30,J)
  201 CONTINUE
      WRITE (6,601)
  601 FORMAT('SHELL LAYER',5X,' SHELL RADIUS  ')
      LIMIT=2*LASHEL
      DO 203 I=1,LIMIT
      WRITE(6,602)I,R2(I,1)
  602 FORMAT(8X,I3,5X,E15.8)
  203 CONTINUE
      RETURN
      END
```

```
C ------------------------------------------------------------------------
C SUBPROGRAM FOR CALCULATING DENSITY GRADIENT AND REFRACTIVE INDEX BET-
C WEEN TWO KNOWN DENSITIES
C A PARABOLIC INCREASING DENSITY PROFILE IS USED FOR APPROXIMATION,NAMELY,
C N=NO(1+R**2/AO**2)
C CONSTANTS NOI,AOI ARE COMPUTED FOR THE EVALUATION OF OMEGA (THE
C SPATIAL FREQUENCY)
C ------------------------------------------------------------------------
      SUBROUTINE DNGR(R,N,LAMDA,NX,RO,NO)
      DIMENSION RDENOR(30,60),N(30,60),R(30,60)
      INTEGER ZSTEP,FLAG
      REAL N,NOI,LAMDA,LAMTA,NDENOR,NO,KA
      NAMELIST /DNGDAT/FRACTN
C ------------------------------------------------------------------------
C          FRACTN---AXIAL DENSITY/1ST SHELL DENSITY
C ------------------------------------------------------------------------
      COMMON /GRIDP/R2(60,60),OMEGA(60,60),LASHEL
      COMMON /DNGRP/RI(60,60),CRIDEN,FLAG(60,60)
      COMMON /DNGRP1/NDENOR(30,60)
      COMMON /ABSOB/AOISQ(60,60),NOI(60,60),LOCX(100,100),LOCY(100,100),
     5*KA(100,100)
      READ(5,DNGDAT)
      WRITE(6,DNGDAT)
      LAMTA=LAMDA*1.0E-4
      CRIDEN=9.1095E-28*3.14159*9.0E20/(LAMTA**2*(4.8032E-10)**2)
C ------------------------------------------------------------------------
C          DENORMALIZING THE RADIUS AND DENSITY
C ------------------------------------------------------------------------
      DO 210 J=1,NX
      DO 211 I=1,LASHEL
      RDENOR(I,J)=R(I,J)*RO
      NDENOR(I,J)=N(I,J)*NO
  211 CONTINUE
  210 CONTINUE
      LIMIT=LASHEL-2
      DO 205 J=1,NX
      SLOPE1=N(2,J)-N(1,J)
      SLOPE3=SLOPE1
      DO 201 I=1,LIMIT
      SLOPE2=N(I+2,J)-N(I+1,J)
      I2M1=2*I-1
      I2=2*I
      I2P1=2*I+1
      IF (SLOPE1.LT.0.0.OR.SLOPE2.LT.0.0) GOTO 5
C------------------------------------------------------------------------
C          TEST FOR DENSITY AT BOUNDARY REGION BETWEEN TWO SHELLS
C------------------------------------------------------------------------
      IF(SLOPE2.LT.SLOPE1.AND.SLOPE3.LT.SLOPE1) GOTO 1
      IF (SLOPE2.LT.SLOPE1) GOTO 2
      GOTO 1
    5 IF(SLOPE1.GT.0.0.AND.SLOPE2.LT.0.0) GOTO 2
      IF (SLOPE1.LT.0.0.AND.SLOPE2.GT.0.0) GOTO 4
      IF (SLOPE2.GT.SLOPE1.AND.SLOPE3.GT.SLOPE1) GOTO 3
      IF (SLOPE2.GT.SLOPE1) GOTO 4
      GOTO 3
C ------------------------------------------------------------------------
C          PARABOLIC INCREASING DENSITY APPROXIMATION
C ------------------------------------------------------------------------
    1 SLOPE3=SLOPE1
      SLOPE1=SLOPE2
      AOISQ(I2,J)=(NDENOR(I,J)*R2(I2P1,J)**2-NDENOR(I+1,J)*R2(I2M1,J
     *)**2)/(NDENOR(I+1,J)-NDENOR(I,J))
      NOI(I2,J)=NDENOR(I,J)/(1+R2(I2M1,J)**2/AOISQ(I2,J))
      OMEGA(I2,J)=SQRT(NOI(I2,J)*9E20/(CRIDEN*AOISQ(I2,J)))
      RI(I2,J)=SQRT(1-NOI(I2,J)/CRIDEN)
      FLAG(I2,J)=1
      AOISQ(I2P1,J)=AOISQ(I2,J)
      NOI(I2P1,J)=NOI(I2,J)
      OMEGA(I2P1,J)=OMEGA(I2,J)
      RI(I2P1,J)=RI(I2,J)
```

```
      FLAG(I2P1,J)=FLAG(I2,J)
      GOTO 201
C ----------------------------------------------------------------------
C          NON-PARABOLIC INCREASING DENSITY APPROXIMATION
C ----------------------------------------------------------------------
    2 SLOPE3=SLOPE1
      SLOPE1=SLOPE2
      TERM1=NDENOR(I+1,J)/R2(I2-1,J)**2
      TERM2=NDENOR(I,J)/R2(I2P1,J)**2
      ADISQ(I2,J)=(NDENOR(I+1,J)-NDENOR(I,J))/(TERM1-TERM2)
      ADISQ(I2P1,J)=ADISQ(I2,J)
      NOI(I2,J)=NDENOR(I,J)/(1-ADISQ(I2,J)**2/R2(I2M1,J)**2)
      NOI(I2P1,J)=NOI(I2,J)
      RI(I2,J)=SQRT(1-NOI(I2,J)/CRIDEN)
      RI(I2P1,J)=RI(I2,J)
      FLAG(I2,J)=2
      FLAG(I2P1,J)=FLAG(I2,J)
      GOTO 201
C ----------------------------------------------------------------------
C          PARABOLIC DECREASING DENSITY APPROXIMATION
C ----------------------------------------------------------------------
    3 SLOPE3=SLOPE1
      SLOPE1=SLOPE2
      ADISQ(I2,J)=(NDENOR(I,J)*R2(I2P1,J)**2-NDENOR(I+1,J)*R2(I2M1,J)**
     *2)/(NDENOR(I,J)-NDENOR(I+1,J))
      NOI(I2,J)=NDENOR(I,J)/(1-R2(I2M1,J)**2/ADISQ(I2,J))
      OMEGA(I2,J)=SQRT(NOI(I,J)*9E20/(CRIDEN*ADISQ(I2,J)))
      RI(I2,J)=SQRT(1-NOI(I2,J)/CRIDEN)
      FLAG(I2,J)=3
      ADISQ(I2P1,J)=ADISQ(I2,J)
      NOI(I2P1,J)=NOI(I2,J)
      OMEGA(I2P1,J)=OMEGA(I2,J)
      RI(I2P1,J)=RI(I2,J)
      FLAG(I2P1,J)=FLAG(I2,J)
      GOTO 201
C ----------------------------------------------------------------------
C          NON-PARABOLIC DECREASING DENSITY APPROXIMATION
C ----------------------------------------------------------------------
    4 SLOPE3=SLOPE1
      SLOPE1=SLOPE2
      T1=NDENOR(I,J)/R2(I2P1,J)**2
      T2=NDENOR(I+1,J)/R2(I2M1,J)**2
      ADISQ(I2,J)=(NDENOR(I+1,J)-NDENOR(I,J))/(T1-T2)
      ADISQ(I2P1,J)=ADISQ(I2,J)
      NOI(I2,J)=NDENOR(I,J)/(1+ADISQ(I2,J)/R2(I2M1,J)**2)
      NOI(I2P1,J)=NOI(I2,J)
      RI(I2,J)=SQRT(1-NOI(I2,J)/CRIDEN)
      RI(I2P1,J)=RI(I2,J)
      FLAG(I2,J)=4
      FLAG(I2P1,J)=FLAG(I2,J)
  201 CONTINUE
      SLOPE1=NDENOR(LASHEL,J)-NDENOR(LASHEL-1,J)
      IF (SLOPE1.LT.0.0) GOTO 7
      TERM1=NDENOR(LASHEL,J)/R2(LASHEL*2-3,J)**2
      TERM2=NDENOR(LASHEL-1,J)/R2(LASHEL*2-1,J)**2
      ADISQ(LASHEL*2-2,J)=(NDENOR(LASHEL,J)-NDENOR(LASHEL-1,J))/
     *(TERM1-TERM2)
      NOI(LASHEL*2-2,J)=NDENOR(LASHEL-1,J)/(1-ADISQ(LASHEL*2-2,J)**
     *2/R2(LASHEL*2-3,J)**2)
      FLAG(LASHEL*2-2,J)=2
      RI(LASHEL*2-2,J)=SQRT(1-NOI(LASHEL*2-2,J)/CRIDEN)
      GOTO 8
    7 T1=NDENOR(LASHEL-1,J)/R2(LASHEL*2-1,J)**2
      T2=NDENOR(LASHEL,J)/R2(2*LASHEL-3,J)**2
      ADISQ(LASHEL*2-2,J)=(NDENOR(LASHEL,J)-NDENOR(LASHEL-1,J))
     */(T1-T2)
      NOI(LASHEL*2-2,J)=NDENOR(LASHEL-1,J)/(1+ADISQ(LASHEL*2-2,J)**2
     */R2(LASHEL*2-1,J)**2)
      RI(LASHEL*2-2,J)=SQRT(1-NOI(LASHEL*2-2,J)/CRIDEN)
      FLAG(LASHEL*2-2,J)=4
```

```
    8 AOISQ(LASHEL*2-1,J)=AOISQ(LASHEL*2-2,J)
      AOISQ(LASHEL*2,J)=AOISQ(LASHEL*2-2,J)
      NOI(LASHEL*2-1,J)=NOI(LASHEL*2-2,J)
      NOI(LASHEL*2,J)=NOI(LASHEL*2-2,J)
      RI(LASHEL*2-1,J)=RI(LASHEL*2-2,J)
      RI(LASHEL*2,J)=RI(LASHEL*2-2,J)
      FLAG(LASHEL*2-1,J)=FLAG(LASHEL*2-2,J)
      FLAG(LASHEL*2,J)=FLAG(LASHEL*2-2,J)
C ---------------------------------------------------------------------------------
C         THE INNERMOST SHELL IS ASSUMED TO HAVE A PARABOLIC DENSITY
C         THE OUTERMOST SHELL IS ASSUMED TO HAVE A NON-PARABOLIC
C         DECREASING DENSITY PROFILE
C ---------------------------------------------------------------------------------
      AOISQ(1,J)=FRACTN*R2(1,J)**2/(1-FRACTN)
      NOI(1,J)=FRACTN*NDENOR(1,J)
      OMEGA(1,J)=SQRT(NOI(1,J)*9E20/(CRIDEN*AOISQ(1,J)))
      RI(1,J)=SQRT(1-NOI(1,J)/CRIDEN)
      FLAG(1,J)=1
      OMEGA(2*LASHEL,J)=0.0
      OMEGA(2*LASHEL-1,J)=0.0
      OMEGA(2*LASHEL-2,J)=0.0
  205 CONTINUE
      WRITE(6,601)
  601 FORMAT('      FLAG      ',5X,'REFRACTIVE INDEX',5X,'      AOI**2
     *',5X,'     NOI      ')
      LASH2=2*LASHEL
      DO 206 I=1,LASH2
      WRITE (6,602) FLAG(I,1),RI(I,1),AOISQ(I,1),NOI(I,1)
  602 FORMAT(8X,I3,9X,E16.8,5X,E15.8,5X,E15.8)
  206 CONTINUE
      RETURN
      END
```

```
C ----------------------------------------------------------------
C SUBROUTINE FOR ENERGY ABSORPTION IN CORR. PLASMA CELLS
C ----------------------------------------------------------------
        SUBROUTINE ENERGY(TIME,STEP,XMIN,NX,XLAS,VLAS,DX,EP,LASPWR,N)
        REAL LASPWR(30,60),EP(30,60),KA,N(30,60),PONDR(30,60),PONDZ(30,60)
        INTEGER TOTRAY,STEP,X,RXMIN,XMIN
        COMMON /LASP/L3,RLO,BMS,NCRIT,TL,DTL,MRMAX
        COMMON /PWRAY/AO
        COMMON /ABSOB/AOISQ(60,60),NOI(60,60),LOCX(100,100),LOCY(100,100)
       *,KA(100,100)
        COMMON /DRAYP/XO(100),YO(100),THETAX(100),THETAY(100),P(100),TOTRA
       *Y,ENI[illegible]),EN(100)
        COMMON /RAYENP/NPTS(100),VSUM(100)
        IF (STEP.EQ.1) GOTO 1
        DO 207 J=1,60
        DO 2[illegible]0
        LASPW[illegible]=0.0
        EP(I,J)=0.0
    208 CONTINUE
    207 CONTINUE
      1 WRITE(6,601)
    601 FORMAT(/' CELL LOCATION ',5X,' POWER IN CELL ',
       *5X,'ENERGY IN CELL ',5X,/,3X,'X',7X,'Y')
C ----------------------------------------------------------------
C          POWER AND ENERGY CALCULATION
C ----------------------------------------------------------------
        DO 204 MARK=1,TOTRAY
        NLAS=60
        JJ=NPTS(MARK)-1
        PIN=P(MARK)/NPTS(MARK)
        ENINI=EN(MARK)/NPTS(MARK)
        LIMIT=NPTS(MARK)
        DO 202 K=2,LIMIT
        POWER=PIN
        ENER=ENINI
        IF (LOCX(MARK,K-1).LT.LOCX(MARK,K)) GOTO 2
        IX=IFIX((LOCX(MARK,K-1)-1)/2.0)+1
        GOTO 3
      2 IX=IFIX(LOCX(MARK,K-1)/2.0)+1
      3 LASPWR(IX,LOCY(MARK,K-1))=LASPWR(IX,LOCY(MARK,K-1))+POWER
        EP(IX,LOCY(MARK,K-1))=EP(IX,LOCY(MARK,K-1))+ENER*(1-EXP(-KA(MARK,K
       *)))
        ENER=ENER*EXP(-KA(MARK,K))
        POWER=POWER*EXP(-KA(MARK,K))
    202 CONTINUE
    204 CONTINUE
        DO 205 J=1,60
        DO 206 I=1,30
        IF (LASPWR(I,J).NE.0.0.OR.EP(I,J).NE.0.0) GOTO 4
        GOTO 206
      4 WRITE(6,602)I,J,LASPWR(I,J),EP(I,J)
    602 FORMAT(1X,I3,5X,I3,8X,E15.8,5X,E15.8,5X,E15.8)
    206 CONTINUE
    205 CONTINUE
        WRITE(12) EP
C ----------------------------------------------------------------
C TO CALCULATE THE RADIAL AND AXIAL PONDEROMOTIVE FORCE
C ----------------------------------------------------------------
        CALL PONDER(LASPWR,VLAS,NLAS,DX,PONDR,PONDZ)
        RETURN
        END
END OF FILE
```

```
C-----------------------------------------------------------------------
C          PROGRAM FOR CALCULATING THE RADIAL AND AXIAL PONDEROMOTIVE FORCE
C          IN INDIVIDUAL CELLS
C-----------------------------------------------------------------------
      SUBROUTINE PONDER(P,VLAS,NLAS,DX,PONDR,PONDZ)
      REAL P(30,60),PONDR(30,60),PONDZ(30,60),LASHEL,NDENOR,AVEI(30,60
     *)
      COMMON /GRIDP/R2(60,60),OMEGA(60,60),LASHEL
      COMMON /DNGRP/RI(60,60),CRIDEN,FLAG(60,60)
      COMMON /DNGRP1/NDENOR(30,60)
C-----------------------------------------------------------------------
C          TO FIND PONDEROMOTIVE FORCES AT THE CELLS AWAY FROM THE
C          BOUNDARY
C-----------------------------------------------------------------------
      DO 201 J=1,NLAS
      DO 202 I=2,30
      AVEI(I,J)=P(I,J)/(3.14159*(R2(2*I,J)**2-R2(2*(I-1),J)**2))
  202 CONTINUE
      AVEI(1,J)=P(1,J)/(3.14159*R2(2,J)**2)
  201 CONTINUE
      NLASM1=NLAS-1
      DO 203 J=2,NLASM1
      DO 204 I=2,29
      PIU=0.5*(AVEI(I+1,J)+AVEI(I,J))
      PIL=0.5*(AVEI(I,J)+AVEI(I-1,J))
      CU=VLAS*SQRT(1-0.5*(NDENOR(I,J)+NDENOR(I+1,J))/CRIDEN)
      CL=VLAS*SQRT(1-0.5*(NDENOR(I,J)+NDENOR(I-1,J))/CRIDEN)
      PILEFT=(AVEI(I,J-1)+AVEI(I,J))*0.5
      PIRITE=(AVEI(I,J+1)+AVEI(I,J))*0.5
      CLEFT=VLAS*SQRT(1-0.5*(NDENOR(I,J-1)+NDENOR(I,J))/CRIDEN)
      CRITE=VLAS*SQRT(1-0.5*(NDENOR(I,J+1)+NDENOR(I,J))/CRIDEN)
      DIDR=(PIU/CU-PIL/CL)/(R2(2*I,J)-R2(2*(I-1),J))
      DIDZ=(PIRITE/CRITE-PILEFT/CLEFT)/DX
      PONDR(I,J)=-0.5*NDENOR(I,J)/CRIDEN*DIDR*1.0E7
      PONDZ(I,J)=-0.5*NDENOR(I,J)/CRIDEN*DIDZ*1.0E7
  204 CONTINUE
C-----------------------------------------------------------------------
C          TO FIND VALUES AT THE INNERMOST LAYER
C          LAYER 1
C-----------------------------------------------------------------------
      PILEFT=(AVEI(1,J-1)+AVEI(1,J))*0.5
      PIRITE=(AVEI(1,J+1)+AVEI(1,J))*0.5
      CLEFT=VLAS*SQRT(1-0.5*(NDENOR(1,J-1)+NDENOR(1,J))/CRIDEN)
      CRITE=VLAS*SQRT(1-0.5*(NDENOR(1,J+1)+NDENOR(1,J))/CRIDEN)
      DIDZ=(PIRITE/CRITE-PILEFT/CLEFT)/DX
      PONDR(1,J)=0.0
      PONDZ(1,J)=-0.5*NDENOR(1,J)/CRIDEN*DIDZ*1.0E7
C-----------------------------------------------------------------------
C          TO FIND POND. FORCES FOR LAYER 30
C-----------------------------------------------------------------------
      PILEFT=(AVEI(30,J-1)+AVEI(30,J))*0.5
      PIRITE=(AVEI(30,J+1)+AVEI(30,J))*0.5
      CLEFT=VLAS*SQRT(1-0.5*(NDENOR(30,J-1)+NDENOR(30,J))/CRIDEN)
      CRITE=VLAS*SQRT(1-0.5*(NDENOR(30,J+1)+NDENOR(30,J))/CRIDEN)
      DIDZ=(PIRITE/CRITE-PILEFT/CLEFT)/DX
      PONDZ(30,J)=-0.5*NDENOR(30,J)/CRIDEN*DIDZ*1.0E7
      PIU=0.5*AVEI(30,J)
      PIL=0.5*(AVEI(30,J)+AVEI(29,J))
      CU=VLAS*SQRT(1-0.5*NDENOR(30,J)/CRIDEN)
      CL=VLAS*SQRT(1-0.5*(NDENOR(30,J)+NDENOR(29,J))/CRIDEN)
      DIDR=(PIU/CU-PIL/CL)/(R2(60,J)-R2(58,J))
      PONDR(30,J)=-0.5*NDENOR(30,J)/CRIDEN*DIDR*1.0E7
  203 CONTINUE
C-----------------------------------------------------------------------
C          POND.FORCES AT COLUMN ONE
C-----------------------------------------------------------------------
      DO 207 I=2,29
      PIU=0.5*(AVEI(I+1,1)+AVEI(I,1))
      PIL=0.5*(AVEI(I,1)+AVEI(I-1,1))
      CU=VLAS*SQRT(1-0.5*(NDENOR(I,1)+NDENOR(I+1,1))/CRIDEN)
```

```
      CL=VLAS*SQRT(1-0.5*(NDENOR(I,1)+NDENOR(I-1,1))/CRIDEN)
      PILEFT=AVEI(I,1)
      PIRITE=(AVEI(I,2)+AVEI(I,1))*0.5
      CLEFT=VLAS*SQRT(1-0.5*NDENOR(I,1)/CRIDEN)
      CRITE=VLAS*SQRT(1-0.5*(NDENOR(I,2)+NDENOR(I,1))/CRIDEN)
      DIDR=(PIU/CU-PIL/CL)/(R2(2*I,1)-R2(2*(I-1),1))
      DIDZ=(PIRITE/CRITE-PILEFT/CLEFT)/DX
      PONDR(I,1)=-0.5*NDENOR(I,1)/CRIDEN*DIDR*1.0E7
      PONDZ(I,1)=-0.5*NDENOR(I,1)/CRIDEN*DIDZ*1.0E7
C-----------------------------------------------------------------------
C         TO FIND FORCES AT COLUMN NLAS
C-----------------------------------------------------------------------
      PIU=0.5*(AVEI(I+1,NLAS)+AVEI(I,NLAS))
      PIL=0.5*(AVEI(I,NLAS)+AVEI(I-1,NLAS))
      CU=VLAS*SQRT(1-0.5*(NDENOR(I,NLAS)+NDENOR(I+1,NLAS))/CRIDEN)
      CL=VLAS*SQRT(1-0.5*(NDENOR(I,NLAS)+NDENOR(I-1,NLAS))/CRIDEN)
      PILEFT=(AVEI(I,NLASM1)+AVEI(I,NLAS))*0.5
      PIRITE=AVEI(I,NLAS)*0.5
      CRITE=VLAS*SQRT(1-0.5*NDENOR(I,NLAS)/CRIDEN)
      CLEFT=VLAS*SQRT(1-0.5*(NDENOR(I,NLAS)+NDENOR(I,NLASM1))/CRIDEN)
      DIDR=(PIU/CU-PIL/CL)/(R2(2*I,NLAS)-R2(2*(I-1),NLAS))
      DIDZ=(PIRITE/CRITE-PILEFT/CLEFT)/DX
      PONDR(I,NLAS)=-0.5*NDENOR(I,NLAS)/CRIDEN*DIDR*1.0E7
      PONDZ(I,NLAS)=-0.5*NDENOR(I,NLAS)/CRIDEN*DIDZ*1.0E7
  207 CONTINUE
C-----------------------------------------------------------------------
C         TO FIND FORCES AT CORNER CELLS
C-----------------------------------------------------------------------
      PIU=0.5*AVEI(30,NLAS)
      PIL=0.5*(AVEI(30,NLAS)+AVEI(29,NLAS))
      CU=VLAS*SQRT(1-0.5*NDENOR(30,NLAS)/CRIDEN)
      CL=VLAS*SQRT(1-0.5*(NDENOR(30,NLAS)+NDENOR(29,NLAS))/CRIDEN)
      PILEFT=(AVEI(30,NLASM1)+AVEI(30,NLAS))*0.5
      PIRITE=AVEI(30,NLAS)*0.5
      CRITE=VLAS*SQRT(1-0.5*NDENOR(30,NLAS)/CRIDEN)
      CLEFT=VLAS*SQRT(1-0.5*(NDENOR(30,NLAS)+NDENOR(30,NLASM1))/CRIDEN)
      DIDR=(PIU/CU-PIL/CL)/(R2(60,NLAS)-R2(58,NLAS))
      DIDZ=(PIRITE/CRITE-PILEFT/CLEFT)/DX
      PONDR(30,NLAS)=-0.5*NDENOR(30,NLAS)/CRIDEN*DIDR*1.0E7
      PONDZ(30,NLAS)=-0.5*NDENOR(30,NLAS)/CRIDEN*DIDZ*1.0E7
C-----------------------------------------------------------------------
C         TO FIND POND.FORCES AT CELL(1,30)
C-----------------------------------------------------------------------
      PILEFT=(AVEI(1,NLASM1)+AVEI(1,NLAS))*0.5
      PIRITE=AVEI(1,NLAS)*0.5
      CRITE=VLAS*SQRT(1-0.5*NDENOR(1,NLAS)/CRIDEN)
      CLEFT=VLAS*SQRT(1-0.5*(NDENOR(1,NLAS)+NDENOR(1,NLASM1))/CRIDEN)
      DIDZ=(PIRITE/CRITE-PILEFT/CLEFT)/DX
      PONDR(1,NLAS)=0.0
      PONDZ(1,NLAS)=-0.5*NDENOR(1,NLAS)/CRIDEN*DIDZ*1.0E7
C-----------------------------------------------------------------------
C         TO FIND POND.FORCES AT CELL(30,1)
C-----------------------------------------------------------------------
      PIU=0.5*AVEI(30,1)
      PIL=0.5*(AVEI(30,1)+AVEI(29,1))
      CU=VLAS*SQRT(1-0.5*NDENOR(30,1)/CRIDEN)
      CL=VLAS*SQRT(1-0.5*(NDENOR(30,1)+NDENOR(29,1))/CRIDEN)
      PILEFT=AVEI(30,1)
      PIRITE=(AVEI(30,2)+AVEI(30,1))*0.5
      CLEFT=VLAS*SQRT(1-0.5*NDENOR(30,1)/CRIDEN)
      CRITE=VLAS*SQRT(1-0.5*(NDENOR(30,2)+NDENOR(30,1))/CRIDEN)
      DIDR=(PIU/CU-PIL/CL)/(R2(60,1)-R2(58,1))
      DIDZ=(PIRITE/CRITE-PILEFT/CLEFT)/DX
      PONDR(30,1)=-0.5*NDENOR(30,1)/CRIDEN*DIDR*1.0E7
      PONDZ(30,1)=-0.5*NDENOR(30,1)/CRIDEN*DIDZ*1.0E7
C-----------------------------------------------------------------------
C         TO FIND POND. FORCES AT CELL(1,1)
C-----------------------------------------------------------------------
      PILEFT=AVEI(1,1)
      PIRITE=(AVEI(1,2)+AVEI(1,1))*0.5
```

```
      CLEFT=VLAS*SQRT(1-0.5*NDENOR(1,1)/CRIDEN)
      CRITE=VLAS*SQRT(1-0.5*(NDENOR(1,2)+NDENOR(1,1))/CRIDEN)
      DIDZ=(PIRITE/CRITE-PILEFT/CLEFT)/DX
      PONDR(1,1)=0.0
      PONDZ(1,1)=-0.5*NDENOR(1,1)/CRIDEN*DIDZ*1.0E7
      WRITE(6,600)
  600 FORMAT(/'CELL LOCATION',5X,'RADIAL POND.FORCE',5X,'AXIAL POND.FORC
     *E',/,3X,'X',5X,'Y')
      DO 205 J=1,NLAS
      DO 206 I=1,30
      IF (PONDR(I,J).EQ.0.0.AND.PONDZ(I,J).EQ.0.0) GOTO 206
      WRITE(6,601)I,J,PONDR(I,J),PONDZ(I,J)
  601 FORMAT(I4,2X,I4,10X,E15.8,6X,E15.8)
  206 CONTINUE
  205 CONTINUE
      WRITE(13) PONDR
      WRITE(14) PONDZ
      RETURN
      END
END OF FILE
```

```
C -----------------------------------------------------------------------
C SUBPROGRAM FOR FINDING THE RAY PATH IN A PLASMA COLUMN WITH A GIVEN
C DENSITY DISTRIBUTION
C GEOMETRICAL APPROACH IS USED
C THE INTERSECTING POINT BETWEEN THE TRAJECTORY AND THE HORIZONTAL
C OR VERTICAL GRID IS COMPUTED
C THE SPATIAL ADVANCEMENT OF THE RAY IS DONE BY INCREASING THE RADIAL
C DISTANCE UNTIL MAXIMUM VALUE IS ACHIEVED
C THE ANALYTICAL EXPRESSIONS USED ARE BASED ON A PARABOLIC DENSITY
C DISTRIBUTION
C BOTH CARTESIAN AND CYLINDRICAL CO-ORDINATES ARE USED
C -----------------------------------------------------------------------
      SUBROUTINE RAYABS(M,NX1,TDENOR,TO,LAMTA,ZL,ZATOM,XLAS,VLAS,TIME,
     *DX,XMIN)
      DOUBLE PRECISION X,Y,VX,VY,DHETAY,DHETAX,DTZ,DTR,Y112,FETA1
      DOUBLE PRECISION FETA2,FETA3,FETA4
      DIMENSION R1(100,100),THETA(100,100),Z(100,100),TE(30,60),
     *TDENOR(30,60)
      REAL MTHETA,NOI,LAMDA,LAMTA,KA
      INTEGER FLAG,TOTRAY,MARK,ZATOM,XMIN,RXMIN
      COMMON /GRIDP/R2(60,60),OMEGA(60,60),LASHEL
      COMMON /DNGRP/RI(60,60),CRIDEN,FLAG(60,60)
      COMMON /DRAYP/XO(100),YO(100),THETAX(100),THETAY(100),P(100),TOTRA
     *Y,ENIN,P1,EN(100)
      COMMON /ABSOB/AOISQ(60,60),NOI(60,60),LOCX(100,100),LOCY(100,100),
     *KA(100,100)
      COMMON /LASP/L3,RLO,BMS,NCRIT,TL,DTL,MRMAX
      COMMON /RAYENP/NPTS(100),VSUM(100)
      EXTERNAL FETA1
      EXTERNAL FETA2
      EXTERNAL FETA3
      EXTERNAL FETA4
C-----------------------------------------------------------------------
C                        INITIALIZATION
C-----------------------------------------------------------------------
      NX=NX1
      C=3.OE10
      LAMDA=LAMTA*1.OE-4
      DO 203 I=1,TOTRAY
  203 VSUM(I)=0.0
C-----------------------------------------------------------------------
C                        DENORMALIZING THE ELECTRON TEMPERATURE
C-----------------------------------------------------------------------
      DO 205 J=1,NX
      DO 206 I=1,M
      TE(I,J)=TDENOR(I,J)*TO
  206 CONTINUE
  205 CONTINUE
      MARK=O
   12 MARK=MARK+1
      IF (MARK.GT.TOTRAY) RETURN
      KA(MARK,1)=0.0
      IF (XO(MARK).EQ.0.0.AND.YO(MARK).EQ.0.0.AND.THETAX(MARK).EQ.0.0.AN
     *D.THETAY(MARK).EQ.0.0) GOTO 110
      X=XO(MARK)
      Y=YO(MARK)
      R1(MARK,1)=SQRT(XO(MARK)*XO(MARK)+YO(MARK)*YO(MARK))
      THETA(MARK,1)=ANGLE(X,Y)
      IF (THETAX(MARK).EQ.0.0.OR.(THETAX(MARK).EQ.0.0.AND.THETAY(MARK).E
     *Q.0.0)) GOTO 13
      DTHETA=ATAN(THETAX(MARK)/THETAY(MARK))
      GOTO 16
   13 VX=0.0
      VY=C*SIN(THETAY(MARK))
      VZ=C*COS(THETAY(MARK))
      GOTO 15
   14 VX=C*SIN(THETAX(MARK))
      VY=0.0
      VZ=C*COS(THETAX(MARK))
      GOTO 15
```

```
   16 MTHETA=SQRT(THETAX(MARK)*THETAX(MARK)+THETAY(MARK)*THETAY(MARK))
      VY=C*SIN(MTHETA)*COS(DTHETA)
      DHETAY=THETAY(MARK)
C----------------------------------------------------------------------
C         VY HAS THE SAME SIGN AS VECTOR THETAY
C----------------------------------------------------------------------
      VY=DSIGN(VY,DHETAY)
      VX=C*SIN(MTHETA)*SIN(DTHETA)
      DHETAX=THETAX(MARK)
C----------------------------------------------------------------------
C         VX HAS THE SAME SIGN AS VECTOR THETAX
C----------------------------------------------------------------------
      VX=DSIGN(VX,DHETAX)
      VZ=C*COS(MTHETA)
   15 Z(MARK,1)=0.0
      DZ=ZL/NX1
      ANGVEL=VX*COS(THETA(MARK,1))-VY*SIN(THETA(MARK,1))
      JJ=1
      J=1
      LAST=2*LASHEL
C----------------------------------------------------------------------
C         SCAN OVER WHICH PLASMA SHELL THE RAY HITS
C----------------------------------------------------------------------
      IF (R1(MARK,1).GT.R2(LAST,1)) GOTO 38
      IF (R1(MARK,1).GT.0.AND.R1(MARK,1).LT.R2(1,1))GOTO 1
      DO 201 II=2,LAST
      IF ((R1(MARK,1).GT.R2(II-1,1).OR.ABS(R1(MARK,1)-R2(II-1,1)).LT.1.0
     *E-3).AND.(R1(MARK,1).LT.R2(II,1).OR.ABS(R2(II,1)-R1(MARK,1)).LT.1.
     *0E-3)) GOTO 2
      GOTO 201
    2 I=II
      GOTO 31
  201 CONTINUE
    1 I=1
   31 LOCX(MARK,1)=I
      LOCY(MARK,1)=J
      IVAR=FLAG(I,J)
C----------------------------------------------------------------------
C         CHOOSE THE APPROPRIATE KIND OF DENSITY PROFILE
C----------------------------------------------------------------------
      GO TO (311,100,33,100),IVAR
  311 TERM1=(X*X+Y*Y)/2-(VX*VX+VY*VY)/(2*OMEGA(I,J)*OMEGA(I,J))
      TERM2=(X*VX+Y*VY)/OMEGA(I,J)
      D=SQRT(TERM1*TERM1+TERM2*TERM2)
      PHI=FI(TERM1,TERM2)
      GOTO 34
  100 ANGMOM=R1(MARK,1)*ANGVEL
      GOTO 34
   33 TERM1=(X*X+Y*Y)/2+(VX*VX+VY*VY)/(2*OMEGA(I,J)*OMEGA(I,J))
      TERM2=(X*VX+Y*VY)/OMEGA(I,J)
      TERM3=R1(MARK,JJ)*R1(MARK,JJ)+(VX*VX+VY*VY)/(OMEGA(I,J)*OMEGA(I,J)
     *)-TERM1
      IF (TERM2.LT.0.0) GOTO 4
      GOTO 3
   34 VR=VX*SIN(THETA(MARK,JJ))+VY*COS(THETA(MARK,JJ))
      IF(ABS(R1(MARK,1)-R2(I,1)).LT.1.0E-3.AND.(VR.GT.0.0)) I=I+1
      IF (VR.LT.0.0) GOTO 4
C----------------------------------------------------------------------
C                   RAY GOES AWAY FROM THE AXIS
C----------------------------------------------------------------------
    3 JJ=JJ+1
      IF (JJ.GT.100) GOTO 37
      IF (I.EQ.1) GOTO 993
      IVAR=FLAG(I,J)
      GO TO(30,101,35,111),IVAR
```

```
C-----------------------------------------------------------------------
C                     RAY CALC.(UP) WITH A PARABOLIC INCREASING DENSITY
C                     PROFILE APPROXIMATION N=NO(1+R**2/A**2)
C-----------------------------------------------------------------------
   30 TERM1=(X*X+Y*Y)/2-(VX*VX+VY*VY)/(2*OMEGA(I,J)*OMEGA(I,J))
      TERM2=(X*VX+Y*VY)/OMEGA(I,J)
      D=SQRT(TERM1*TERM1+TERM2*TERM2)
      PHI=FI(TERM1,TERM2)
      RMAX=BIGR(R1(MARK,JJ-1),VX,VY,D,I,J)
      R1(MARK,JJ)=R2(I,J)
      IF (R1(MARK,JJ).GT.RMAX) GOTO 5
      TR=RTIME(R1(MARK,JJ-1),R1(MARK,JJ),VX,VY,I,J,PHI,D)
      VZ=DSQRT(RI(I,J)*RI(I,J)*9.OD20-VX*VX-VY*VY-OMEGA(I,J)*OMEGA(I,J)
     **R1(MARK,JJ-1)*R1(MARK,JJ-1)*1.ODO)
      Z(MARK,JJ)=VZ*TR+Z(MARK,JJ-1)
      IF (Z(MARK,JJ).GT.J*DZ) GOTO 6
      VSUM(MARK)=VSUM(MARK)+VZ
      CALL ABSORB(MARK,R1(MARK,JJ-1),Z(MARK,JJ-1),DZ,TR,VX,VY,PHI,LAMDA,
     *TE,D,I,J,JJ,ZATOM)
      VXO=VX
      VYO=VY
      VX=XVEL(X,VXO,TR,I,J)
      VY=YVEL(Y,VYO,TR,I,J)
      X=XCOOR(X,VXO,TR,I,J)
      Y=YCOOR(Y,VYO,TR,I,J)
      GOTO 36
C-----------------------------------------------------------------------
C                     RAY CALC.(UP) WITH A PARABOLIC DECREASING DENSITY
C                     PROFILE APPROXIMATION N=NO(1-R**2/A**2)
C-----------------------------------------------------------------------
   35 R1(MARK,JJ)=R2(I,J)
      TERM1=(X*X+Y*Y)/2+(VX*VX+VY*VY)/(2*OMEGA(I,J)*OMEGA(I,J))
      TERM2=(X*VX+Y*VY)/OMEGA(I,J)
      TERM3=R1(MARK,JJ)*R1(MARK,JJ)+(VX*VX+VY*VY)/(OMEGA(I,J)*OMEGA(I,J)
     *)-TERM1
      TERM4=TERM1-(VX*VX+VY*VY)/(OMEGA(I,J)*OMEGA(I,J))
      NC=O
      TR=HTIME(TERM1,TERM2,TERM3,I,J,NC)
      IF (TR.LT.O.O.OR.NC.EQ.1)GOTO 39
      VZ=DSQRT(RI(I,J)*RI(I,J)*9D20-VX*VX-VY*VY+OMEGA(I,J)*OMEGA(I,J)
     **R1(MARK,JJ-1)*R1(MARK,JJ-1)*1.ODO)
      Z(MARK,JJ)=VZ*TR+Z(MARK,JJ-1)
      IF(Z(MARK,JJ).GT.J*DZ) GOTO 11
      VSUM(MARK)=VSUM(MARK)+VZ
      CALL ABSOB1(MARK,TR,VX,VY,VZ,LAMDA,TE,
     *TERM1,TERM2,I,J,JJ,ZATOM,Z(MARK,JJ-1),DZ)
      VXO=VX
      VYO=VY
      VX=HXVEL(X,VXO,TR,I,J)
      VY=HYVEL(Y,VYO,TR,I,J)
      X=HXCOOR(X,VXO,TR,I,J)
      Y=HYCOOR(Y,VYO,TR,I,J)
   36 THETA(MARK,JJ)=ANGLE(X,Y)
      VR=VX*SIN(THETA(MARK,JJ))+VY*COS(THETA(MARK,JJ))
      ANGVEL=VX*COS(THETA(MARK,JJ))-VY*SIN(THETA(MARK,JJ))
      ANGMOM=R1(MARK,JJ)*ANGVEL
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      I=I+1
      IF (I.GT.LAST) GOTO 32
      GOTO 3
C-----------------------------------------------------------------------
C                     RAY CALC.(UP) WITH A NON-PARABOLIC INCREASING
C                     DENSITY PROFILE N=NO(1-A**2/R**2)
C                     CONST>0.0 OR <0.0, VR>0.0
C-----------------------------------------------------------------------
  101 CONST=(ANGMOM*ANGMOM-9E20*NOI(I,J)*AOISQ(I,J)/CRIDEN)
      IF(CONST.LT.O.O) GOTO 105
  112 R1(MARK,JJ)=R2(I,J)
      TR=TIMEPP(R1(MARK,JJ-1),R1(MARK,JJ),VR,CONST)
      VZ=SQRT((RI(I,J)*3E10)**2-VR*VR-(ANGMOM/R1(MARK,JJ-1))**2)
```

```
      Z(MARK,JJ)=VZ*TR+Z(MARK,JJ-1)
      IF (Z(MARK,JJ).GT.J*DZ) GOTO 102
      VSUM(MARK)=VSUM(MARK)+VZ
      CALL ABOB2P(MARK,R1(MARK,JJ-1),Z(MARK,JJ-1),DZ,TR,VR,CONST,LAMDA,
     *TE,I,J,JJ,ZATOM)
      DTR=TR*1.0D0
      IF (IVAR.EQ.2) GOTO 1121
      CALL DQG32(0.D0,DTR,FETA3,Y112)
      GOTO 1122
 1121 CALL DQG32(0.D0,DTR,FETA1,Y112)
 1122 THETA(MARK,JJ)=THETA(MARK,JJ-1)+Y112*ANGMOM
      VR=SQRT(VR*VR-CONST*(1/(R1(MARK,JJ)*R1(MARK,JJ))-1/(R1(MARK,JJ-1)*
     *R1(MARK,JJ-1))))
      VT=ANGMOM/R1(MARK,JJ)
      VX=VR*SIN(THETA(MARK,JJ))+VT*COS(THETA(MARK,JJ))
      VY=VR*COS(THETA(MARK,JJ))-VT*SIN(THETA(MARK,JJ))
      X=R1(MARK,JJ)*SIN(THETA(MARK,JJ))
      Y=R1(MARK,JJ)*COS(THETA(MARK,JJ))
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      I=I+1
      IF (I.GT.LAST) GOTO 32
      GOTO 3
C-----------------------------------------------------------------------
C          RAY CALC.(UP) FOR A NON-PARABOLIC DECREASING PROFILE
C          N=NO(1+A**2/R**2)
C-----------------------------------------------------------------------
  111 CONST=(ANGMOM*ANGMOM+9E20*NOI(I,J)*AOISQ(I,J)/CRIDEN)
      GOTO 112
   32 WRITE(6,604)
  604 FORMAT(/'***** THE RAY GOES OFF THE PRESCRIBED PLASMA SHELLS FOR
     *BEAM CALCULATION *****'/)
      GOTO 666
   39 JJ=JJ-1
      WRITE(6,606)
  606 FORMAT(/'***** RAY DOES NOT PENETRATE THE PLASMA *****'/)
      GOTO 666
   37 JJ=JJ-1
      WRITE(6,610)
  610 FORMAT(/'***** ATTENTION---STORAGE FOR RAY LOCATION OVERFLOW *****
     *,INCREASE STORAGE '/)
      GOTO 666
   38 WRITE(6,611)
  611 FORMAT(/'*****ATTENTION---RAY IS OUT OF THE PLASMA SHELLS RANGE***
     ***'/)
      GOTO 12
C-----------------------------------------------------------------------
C          RAY GOES TOWARDS INNER SHELLS
C-----------------------------------------------------------------------
    4 IF (I.EQ.1) GOTO 9
      JJ=JJ+1
      IF (JJ.GT.100) GOTO 37
      IVAR=FLAG(I,J)
      GO TO (42,103,41,113),IVAR
C-----------------------------------------------------------------------
C          RAY CALC.(UP) WITH A PARABOLIC INCREASING DENSITY
C          PROFILE APPROXIMATION N=NO(1+R**2/A**2)
C-----------------------------------------------------------------------
   42 TERM1=(X*X+Y*Y)/2-(VX*VX+VY*VY)/(2*OMEGA(I,J)*OMEGA(I,J))
      TERM2=(X*VX+Y*VY)/OMEGA(I,J)
      D=SQRT(TERM1*TERM1+TERM2*TERM2)
      PHI=FI(TERM1,TERM2)
      RMIN=SMALR(R1(MARK,JJ-1),VX,VY,D,I,J)
      R1(MARK,JJ)=R2(I-1,J)
      IF (R1(MARK,JJ).LE.RMIN) GOTO 7
      TR=RTIME(R1(MARK,JJ-1),R1(MARK,JJ),VX,VY,I,J,PHI,D)
      VZ=DSQRT(RI(I,J)**2*9D20-VX*VX-VY*VY-OMEGA(I,J)*OMEGA(I,J)
     **R1(MARK,JJ-1)*R1(MARK,JJ-1)*1.0D0)
      Z(MARK,JJ)=VZ*TR+Z(MARK,JJ-1)
      IF (Z(MARK,JJ).GT.J*DZ) GOTO 6
```

```
      VSUM(MARK)=VSUM(MARK)+VZ
      CALL ABSORB(MARK,R1(MARK,JJ-1),Z(MARK,JJ-1),DZ,TR,VX,VY,PHI,LAMDA,
     *TE,D,I,J,JJ,ZATOM)
      VXO=VX
      VYO=VY
      VX=XVEL(X,VXO,TR,I,J)
      VY=YVEL(Y,VYO,TR,I,J)
      X=XCOOR(X,VXO,TR,I,J)
      Y=YCOOR(Y,VYO,TR,I,J)
      GOTO 40
C-----------------------------------------------------------------------
C         RAY CALC.(UP) WITH A PARABOLIC DECREASING DENSITY
C         PROFILE APPROXIMATION N=NO(1-R**2/A**2)
C-----------------------------------------------------------------------
   41 R1(MARK,JJ)=R2(I-1,J)
      TERM1=(X*X+Y*Y)/2+(VX*VX+VY*VY)/(2*OMEGA(I,J)*OMEGA(I,J))
      TERM2=(X*VX+Y*VY)/OMEGA(I,J)
      IF (TERM1.LT.TERM2) GOTO 39
      TERM3=R1(MARK,JJ)**2+(VX*VX+VY*VY)/OMEGA(I,J)**2-TERM1
      TERM4=TERM1-(VX*VX+VY*VY)/(OMEGA(I,J)*OMEGA(I,J))
      RMIN=TERM1-(VX*VX+VY*VY)/OMEGA(I,J)**2+SQRT(TERM1**2-TERM2**2)
      IF (RMIN.LT.0.0) GOTO 39
      RMIN=SQRT(RMIN)
      IF (R1(MARK,JJ).LT.RMIN) GOTO 43
      NC=0
      TR=HTIME(TERM1,TERM2,TERM3,I,J,NC)
      IF (TR.LT.0.0.OR.NC.EQ.1) GOTO 39
      VZ=DSQRT(RI(I,J)**2*9D20-VX*VX-VY*VY+OMEGA(I,J)*OMEGA(I,J)
     **R1(MARK,JJ-1)*R1(MARK,JJ-1)*1.0D0)
      Z(MARK,JJ)=VZ*TR+Z(MARK,JJ-1)
      IF(Z(MARK,JJ).GT.J*DZ) GOTO 11
      VSUM(MARK)=VSUM(MARK)+VZ
      CALL ABSOB1(MARK,TR,VX,VY,VZ,LAMDA,TE,
     *TERM1,TERM2,I,J,JJ,ZATOM,Z(MARK,JJ-1),DZ)
      VXO=VX
      VYO=VY
      VX=HXVEL(X,VXO,TR,I,J)
      VY=HYVEL(Y,VYO,TR,I,J)
      X=HXCOOR(X,VXO,TR,I,J)
      Y=HYCOOR(Y,VYO,TR,I,J)
   40 THETA(MARK,JJ)=ANGLE(X,Y)
      VR=VX*SIN(THETA(MARK,JJ))+VY*COS(THETA(MARK,JJ))
      ANGVEL=VX*COS(THETA(MARK,JJ))-VY*SIN(THETA(MARK,JJ))
      ANGMOM=R1(MARK,JJ)*ANGVEL
      I=I-1
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      GOTO 4
C-----------------------------------------------------------------------
C         RAY CALC.(DOWN) WITH NON-PARABOLIC INCREASING PROFILE
C         N=NO(1-A**2/R**2)
C         VR<0.0,CONST>0.0 OR<0.0
C-----------------------------------------------------------------------
  103 CONST=(ANGMOM*ANGMOM-9E20*NOI(I,J)*AOISQ(I,J)/CRIDEN)
      IF (CONST.LT.0.0) GOTO 107
  104 RMIN=SQRT((R1(MARK,JJ-1)**2*CONST)/((R1(MARK,JJ-1)*VR)**2+CONST))
      R1(MARK,JJ)=R2(I-1,J)
      IF (R1(MARK,JJ).LT.RMIN) GOTO 106
      TR=TIMEPN(R1(MARK,JJ-1),R1(MARK,JJ),VR,CONST)
      VZ=SQRT((RI(I,J)*3E10)**2-VR*VR-(ANGMOM/R1(MARK,JJ-1))**2)
      Z(MARK,JJ)=VZ*TR+Z(MARK,JJ-1)
      IF (Z(MARK,JJ).GT.J*DZ)GOTO 1021
      VSUM(MARK)=VSUM(MARK)+VZ
      CALL ABOB2P(MARK,R1(MARK,JJ-1),Z(MARK,JJ-1),DZ,TR,VR,CONST,LAMDA,
     *TE,I,J,JJ,ZATOM)
      DTR=TR*1.0D0
      IF (IVAR.EQ.2) GOTO 1041
      CALL DQG32(0.D0,DTR,FETA4,Y112)
      GOTO 1042
 1041 CALL DQG32(0.D0,DTR,FETA2,Y112)
 1042 THETA(MARK,JJ)=THETA(MARK,JJ-1)+Y112*ANGMOM
```

```
      VR=-SQRT(VR*VR-CONST*(1/R1(MARK,JJ)**2-1/R1(MARK,JJ-1)**2))
      VT=ANGMOM/R1(MARK,JJ)
      VX=VR*SIN(THETA(MARK,JJ))+VT*COS(THETA(MARK,JJ))
      VY=VR*COS(THETA(MARK,JJ))-VT*SIN(THETA(MARK,JJ))
      X=R1(MARK,JJ)*SIN(THETA(MARK,JJ))
      Y=R1(MARK,JJ)*COS(THETA(MARK,JJ))
      I=I-1
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      GOTO 4
C-----------------------------------------------------------------------
C          RAY CALC.(UP) WITH NON-PARABOLIC DECREASING PROFILE
C          N=NO(1+A**2/R**2)
C-----------------------------------------------------------------------
  113 CONST=(ANGMOM*ANGMOM+9E20*NOI(I,J)*AOISQ(I,J)/CRIDEN)
      GOTO 104
C-----------------------------------------------------------------------
C          RAY REACHES MINIMUM/MAXIMUM
C          N=NO(1+R**2/A**2)
C-----------------------------------------------------------------------
    7 R1(MARK,JJ)=RMIN
      GOTO 8
    5 R1(MARK,JJ)=RMAX
    8 TR=RTIME(R1(MARK,JJ-1),R1(MARK,JJ),VX,VY,I,J,PHI,D)
      VZ=DSQRT(RI(I,J)**2*9D20-VX*VX-VY*VY-OMEGA(I,J)*OMEGA(I,J)
     **R1(MARK,JJ-1)**2*1.0D0)
      Z(MARK,JJ)=VZ*TR+Z(MARK,JJ-1)
      IF (Z(MARK,JJ).GT.J*DZ) GOTO 6
      VSUM(MARK)=VSUM(MARK)+VZ
      CALL ABSORB(MARK,R1(MARK,JJ-1),Z(MARK,JJ-1),TR,VX,VY,PHI,LAMDA,
     *TE,D,I,J,JJ,ZATOM)
      VXO=VX
      VYO=VY
      VX=XVEL(X,VXO,TR,I,J)
      VY=YVEL(Y,VYO,TR,I,J)
      X=XCOOR(X,VXO,TR,I,J)
      Y=YCOOR(Y,VYO,TR,I,J)
      THETA(MARK,JJ)=ANGLE(X,Y)
      VR=0.0
      ANGVEL=VX*COS(THETA(MARK,JJ))-VY*SIN(THETA(MARK,JJ))
      ANGMOM=R1(MARK,JJ)*ANGVEL
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      IF (R1(MARK,JJ-1).GT.R1(MARK,JJ)) GOTO 3
      GOTO 4
C -----------------------------------------------------------------------
C          N=NO(1-R**2/A**2)   FIND MIN. RADIUS
C -----------------------------------------------------------------------
   43 R1(MARK,JJ)=RMIN
      TR=0.25*ALOG(ABS(TERM1+ABS(TERM2))/(TERM1-ABS(TERM2)))/OMEGA(I,J)
      VZ=DSQRT(RI(I,J)*RI(I,J)*9D20-VX*VX-VY*VY+OMEGA(I,J)*OMEGA(I,J)
     **R1(MARK,JJ-1)*R1(MARK,JJ-1)*1.0D0)
      Z(MARK,JJ)=VZ*TR+Z(MARK,JJ-1)
      IF(Z(MARK,JJ).GT.J*DZ) GOTO 11
      VSUM(MARK)=VSUM(MARK)+VZ
      CALL ABSOB1(MARK,TR,VX,VY,VZ,LAMDA,TE,
     *TERM1,TERM2,I,J,JJ,ZATOM,Z(MARK,JJ-1),DZ)
      VXO=VX
      VYO=VY
      VX=HXVEL(X,VXO,TR,I,J)
      VY=HYVEL(Y,VYO,TR,I,J)
      X=HXCOOR(X,VXO,TR,I,J)
      Y=HYCOOR(Y,VYO,TR,I,J)
      THETA(MARK,JJ)=ANGLE(X,Y)
      VR=VX*SIN(THETA(MARK,JJ))+VY*COS(THETA(MARK,JJ))
      ANGVEL=VX*COS(THETA(MARK,JJ))-VY*SIN(THETA(MARK,JJ))
      ANGMOM=R1(MARK,JJ)*ANGVEL
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      GOTO 3
```

```
      C-----------------------------------------------------------------------
      C          RAY HITS AT VERTICAL BOUNDARY
      C          N=NO(1+R**2/A**2)
      C -----------------------------------------------------------------------
        6 Z(MARK,JJ)=J*DZ
          TZ=(Z(MARK,JJ)-Z(MARK,JJ-1))/VZ
          R1(MARK,JJ)=DSQRT(R1(MARK,JJ-1)**2/2.DO+(VX*VX+VY*VY)/(2*OMEGA(I,J
         *)*OMEGA(I,J))+D*1.ODO*SIN(2*OMEGA(I,J)*TZ+PHI))
          CALL ABSORB(MARK,R1(MARK,JJ-1),Z(MARK,JJ-1),DZ,TZ,VX,VY,PHI,LAMDA,
         *TE,D,I,J,JJ,ZATOM)
          VXO=VX
          VYO=VY
          VX=XVEL(X,VXO,TZ,I,J)
          VY=YVEL(Y,VYO,TZ,I,J)
          X=XCOOR(X,VXO,TZ,I,J)
          Y=YCOOR(Y,VYO,TZ,I,J)
          THETA(MARK,JJ)=ANGLE(X,Y)
          VR=VX*SIN(THETA(MARK,JJ))+VY*COS(THETA(MARK,JJ))
          ANGVEL=VX*COS(THETA(MARK,JJ))-VY*SIN(THETA(MARK,JJ))
          ANGMOM=R1(MARK,JJ)*ANGVEL
          VSUM(MARK)=VSUM(MARK)+VZ
          J=J+1
          LOCX(MARK,JJ)=I
          LOCY(MARK,JJ)=J
          IF (J.GT.NX) GOTO 666
          IF (VR.LT.O.O)GOTO 4
          GOTO 3
      C -----------------------------------------------------------------------
      C          N=NO(1-R**2/A**2)
      C -----------------------------------------------------------------------
       11 Z(MARK,JJ)=J*DZ
          TZ=(Z(MARK,JJ)-Z(MARK,JJ-1))/VZ
          ARG=SINH(2*OMEGA(I,J)*TZ)
          R1(MARK,JJ)=DSQRT(TERM1*(1+DSQRT(1+1.ODO*ARG*ARG))+TERM2*ARG-(VX**
         *2+VY**2)/OMEGA(I,J)**2)
          CALL ABSOB1(MARK,TZ,VX,VY,VZ,LAMDA,TE,
         *TERM1,TERM2,I,J,JJ,ZATOM,Z(MARK,JJ-1),DZ)
          VSUM(MARK)=VSUM(MARK)+VZ
          VXO=VX
          VYO=VY
          VX=HXVEL(X,VXO,TZ,I,J)
          VY=HYVEL(Y,VYO,TZ,I,J)
          X=HXCOOR(X,VXO,TZ,I,J)
          Y=HYCOOR(Y,VYO,TZ,I,J)
          THETA(MARK,JJ)=ANGLE(X,Y)
          VR=VX*SIN(THETA(MARK,JJ))+VY*COS(THETA(MARK,JJ))
          ANGVEL=VX*COS(THETA(MARK,JJ))-VY*SIN(THETA(MARK,JJ))
          ANGMOM=R1(MARK,JJ)*ANGVEL
          J=J+1
          LOCX(MARK,JJ)=I
          LOCY(MARK,JJ)=J
          IF (J.GT.NX) GOTO 666
          IF (VR.LT.O.O)GOTO 4
          GOTO 3
      C -----------------------------------------------------------------------
      C          N=NO(1-A**2/R**2), CONST<O.O, VR>O.O,
      C-----------------------------------------------------------------------
      109 Z(MARK,JJ)=J*DZ
          TZ=(Z(MARK,JJ)-Z(MARK,JJ-1))/VZ
          TEMP1=(R1(MARK,JJ-1)*VR)**2-ABS(CONST)
          TEMP=(TZ+R1(MARK,JJ-1)*R1(MARK,JJ-1)*R1(MARK,JJ-1)*VR/TEMP1)**2
          TEMP=TEMP1*TEMP1*TEMP/R1(MARK,JJ-1)**2
          TEMP=TEMP-ABS(CONST)*R1(MARK,JJ-1)**2
          R1(MARK,JJ)=SQRT(TEMP/TEMP1)
          CALL ABOB2N(MARK,R1(MARK,JJ-1),Z(MARK,JJ-1),DZ,TZ,VR,CONST,LAMDA,
         *TE,I,J,JJ,ZATOM)
          DTZ=TZ*1.ODO
          CALL DQG32(O.DO,DTZ,FETA1,Y112)
          THETA(MARK,JJ)=THETA(MARK,JJ-1)+Y112*ANGMOM
          VSUM(MARK)=VSUM(MARK)+VZ
          VR=SQRT(VR*VR+ABS(CONST)*(1/R1(MARK,JJ)**2-1/R1(MARK,JJ-1)**2))
```

```
      VT=ANGMOM/R1(MARK,JJ)
      VX=VR*SIN(THETA(MARK,JJ))+VT*COS(THETA(MARK,JJ))
      VY=VR*COS(THETA(MARK,JJ))-VT*SIN(THETA(MARK,JJ))
      X=R1(MARK,JJ)*SIN(THETA(MARK,JJ))
      Y=R1(MARK,JJ)*COS(THETA(MARK,JJ))
      J=J+1
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      IF (J.GT.NX) GOTO 666
      GOTO 3
C ----------------------------------------------------------------
C          N=NO(1-A**2/R**2), CONST<0.0, VR<0.0,
C ----------------------------------------------------------------
 1091 Z(MARK,JJ)=J*DZ
      TZ=(Z(MARK,JJ)-Z(MARK,JJ-1))/VZ
      TEMP1=(R1(MARK,JJ-1)*VR)**2-ABS(CONST)
      TEMP=(TZ-R1(MARK,JJ-1)*R1(MARK,JJ-1)*R1(MARK,JJ-1)*ABS([illegible])/TEMP1)*
     **2
      TEMP=TEMP1*TEMP1*TEMP/R1(MARK,JJ-1)**2
      TEMP=TEMP-ABS(CONST)*R1(MARK,JJ-1)**2
      R1(MARK,JJ)=SQRT(TEMP/TEMP1)
      CALL ABOB2N(MARK,R1(MARK,JJ-1),Z(MARK,JJ-1),DZ,TZ,VR,CONST,LAMDA,
     *TE,I,J,JJ,ZATOM)
      DTZ=TZ*1.0D0
      CALL DQG32(0.D0,DTZ,FETA2,Y112)
      THETA(MARK,JJ)=THETA(MARK,JJ-1)+Y112*ANGMOM
      VR=-SQRT(VR*VR+ABS(CONST)*(1/R1(MARK,JJ)**2-1/R1(MARK,JJ-1)**2))
      VT=ANGMOM/R1(MARK,JJ)
      VX=VR*SIN(THETA(MARK,JJ))+VT*COS(THETA(MARK,JJ))
      VY=VR*COS(THETA(MARK,JJ))-VT*SIN(THETA(MARK,JJ))
      X=R1(MARK,JJ)*SIN(THETA(MARK,JJ))
      Y=R1(MARK,JJ)*COS(THETA(MARK,JJ))
      J=J+1
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      VSUM(MARK)=VSUM(MARK)+VZ
      IF (J.GT.NX) GOTO 666
      GOTO 4
C ----------------------------------------------------------------
C          N=NO(1-A**2/R**2), OR N=NO(1+A**2/R**2), CONST>0.0, VR>0.0
C ----------------------------------------------------------------
  102 Z(MARK,JJ)=J*DZ
      TZ=(Z(MARK,JJ)-Z(MARK,JJ-1))/VZ
      TEMP1=(R1(MARK,JJ-1)*VR)**2+CONST
      TEMP=(TZ+R1(MARK,JJ-1)*R1(MARK,JJ-1)*R1(MARK,JJ-1)*VR/TEMP1)**2
      TEMP=TEMP1*TEMP1*TEMP/R1(MARK,JJ-1)**2
      TEMP=TEMP+CONST*R1(MARK,JJ-1)*R1(MARK,JJ-1)
      R1(MARK,JJ)=SQRT(TEMP/TEMP1)
      CALL ABOB2P(MARK,R1(MARK,JJ-1),Z(MARK,JJ-1),DZ,TZ,VR,CONST,LAMDA,
     *TE,I,J,JJ,ZATOM)
      THETA(MARK,JJ)=THETA(MARK,JJ-1)+ANGMOM*TR/R1(MARK,JJ)**2
      VSUM(MARK)=VSUM(MARK)+VZ
      VR=SQRT(VR*VR-CONST*(1/R1(MARK,JJ)**2-1/R1(MARK,JJ-1)**2))
      VT=ANGMOM/R1(MARK,JJ)
      VX=VR*SIN(THETA(MARK,JJ))+VT*COS(THETA(MARK,JJ))
      VY=VR*COS(THETA(MARK,JJ))-VT*SIN(THETA(MARK,JJ))
      X=R1(MARK,JJ)*SIN(THETA(MARK,JJ))
      Y=R1(MARK,JJ)*COS(THETA(MARK,JJ))
      J=J+1
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      IF (J.GT.NX) GOTO 666
      GOTO 3
C ----------------------------------------------------------------
C          N=NO(1-A**2/R**2), CONST>0.0, VR<0.0
C ----------------------------------------------------------------
 1021 Z(MARK,JJ)=J*DZ
      TZ=(Z(MARK,JJ)-Z(MARK,JJ-1))/VZ
      TEMP1=(R1(MARK,JJ-1)*VR)**2+CONST
      TEMP=(TZ-R1(MARK,JJ-1)*R1(MARK,JJ-1)*R1(MARK,JJ-1)*ABS(VR)/TEMP1)*
     **2
```

```
      TEMP=TEMP1*TEMP1*TEMP/R1(MARK,JJ-1)**2
      TEMP=TEMP+CONST*R1(MARK,JJ-1)*R1(MARK,JJ-1)
      R1(MARK,JJ)=SQRT(TEMP/TEMP1)
      CALL ABOB2P(MARK,R1(MARK,JJ-1),Z(MARK,JJ-1),DZ,TZ,VR,CONST,LAMDA,
     *TE,I,J,JJ,ZATOM)
      DTZ=TZ*1.0D0
      CALL DQG32(0.D0,DTZ,FETA3,Y112)
      THETA(MARK,JJ)=THETA(MARK,JJ-1)+ANGMOM*Y112
      VSUM(MARK)=VSUM(MARK)+VZ
      VR=-SQRT(VR**2-CONST*(1/R1(MARK,JJ)**2-1/R1(MARK,JJ-1)**2))
      VT=ANGMOM/R1(MARK,JJ)
      VX=VR*SIN(THETA(MARK,JJ))+VT*COS(THETA(MARK,JJ))
      VY=VR*COS(THETA(MARK,JJ))-VT*SIN(THETA(MARK,JJ))
      X=R1(MARK,JJ)*SIN(THETA(MARK,JJ))
      Y=R1(MARK,JJ)*COS(THETA(MARK,JJ))
      J=J+1
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      IF (J.GT.NX) GOTO 666
      GOTO 4
C---------------------------------------------------------------------
C         FIND MAX./MIN.RADIUS FOR THE PROFILE
C         N=NO(1-A**2/R**2)
C
C         CASE 1 : CONST<0.0 OR 0.0, VR>0.0, FIND MAX. RADIUS
C ---------------------------------------------------------------------
  105 IF (R1(MARK,JJ-1).LE.R2(I,J)) VR=ABS(VR)
      RMAX=R1(MARK,JJ-1)**2*ABS(CONST)/(ABS(CONST)-(R1(MARK,JJ-1)*
     *VR)**2)
      IF (RMAX.LT.0.0) GOTO 1052
      RMAX=SQRT(RMAX)
      GOTO 1051
 1052 RMAX=1.0E6
 1051 R1(MARK,JJ)=R2(I,J)
      IF (R1(MARK,JJ).GT.RMAX) GOTO 108
      TR=TIMENP(R1(MARK,JJ-1),R1(MARK,JJ),VR,CONST)
      VZ=SQRT((RI(I,J)*3E10)**2-VR*VR-(ANGMOM/R1(MARK,JJ-1))**2)
      Z(MARK,JJ)=Z(MARK,JJ-1)+VZ*TR
      IF (Z(MARK,JJ).GT.J*DZ) GOTO 109
      CALL ABOB2N(MARK,R1(MARK,JJ-1),Z(MARK,JJ-1),DZ,TR,VR,CONST,LAMDA,
     *TE,I,J,JJ,ZATOM)
      DTR=TR*1.0D0
      CALL DQG32(0.D0,DTR,FETA1,Y112)
      VSUM(MARK)=VSUM(MARK)+VZ
      THETA(MARK,JJ)=THETA(MARK,JJ-1)+Y112*ANGMOM
      VR=SQRT(VR*VR+ABS(CONST)*(1/R1(MARK,JJ)**2-1/R1(MARK,JJ-1)**2))
      VT=ANGMOM/R1(MARK,JJ)
      VX=VR*SIN(THETA(MARK,JJ))+VT*COS(THETA(MARK,JJ))
      VY=VR*COS(THETA(MARK,JJ))-VT*SIN(THETA(MARK,JJ))
      X=R1(MARK,JJ)*SIN(THETA(MARK,JJ))
      Y=R1(MARK,JJ)*COS(THETA(MARK,JJ))
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      I=I+1
      IF (I.GT.LAST) GOTO 32
      GOTO 3
C ---------------------------------------------------------------------
C     CASE 2 : N=NO(1-A**2/R**2), CONST<0.0, VR>0.0, FIND MAX. RADIUS
C ---------------------------------------------------------------------
  108 R1(MARK,JJ)=RMAX
      TR=TIMENP(R1(MARK,JJ-1),R1(MARK,JJ),VR,CONST)
      VZ=SQRT((RI(I,J)*3E10)**2-VR*VR-(ANGMOM/R1(MARK,JJ-1))**2)
      Z(MARK,JJ)=VZ*TR+Z(MARK,JJ-1)
      IF (Z(MARK,JJ).GT.J*DZ) GOTO 109
      CALL ABOB2N(MARK,R1(MARK,JJ-1),Z(MARK,JJ-1),DZ,TR,VR,CONST,LAMDA,
     *TE,I,J,JJ,ZATOM)
      DTR=TR*1.0D0
      CALL DQG32(0.D0,DTR,FETA1,Y112)
      VSUM(MARK)=VSUM(MARK)+VZ
      THETA(MARK,JJ)=THETA(MARK,JJ-1)+Y112*ANGMOM
      VR=0.0
```

```
      VT=ANGMOM/R1(MARK,JJ)
      VX=VR*SIN(THETA(MARK,JJ))+VT*COS(THETA(MARK,JJ))
      VY=VR*COS(THETA(MARK,JJ))-VT*SIN(THETA(MARK,JJ))
      X=R1(MARK,JJ)*SIN(THETA(MARK,JJ))
      Y=R1(MARK,JJ)*COS(THETA(MARK,JJ))
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      GOTO 4
C ---------------------------------------------------------------------
C     CASE 3: N=NO(1-A**2/R**2), CONST<0.0, VR<0.0, FIND MIN. RADIUS
C ---------------------------------------------------------------------
  107 RMIN=ABS(ANGMOM/(RI(I,J)*3E10))
      R1(MARK,JJ)=R2(I-1,J)
      IF (R1(MARK,JJ).LT.RMIN) GOTO 120
      TR=TIMENN(R1(MARK,JJ-1),R1(MARK,JJ),VR,CONST)
      VZ=SQRT((RI(I,J)*3E10)**2-VR*VR-(ANGMOM/R1(MARK,JJ-1))**2)
      Z(MARK,JJ)=VZ*TR+Z(MARK,JJ-1)
      IF (Z(MARK,JJ).GT.J*DZ) GOTO 1091
      CALL ABOB2N(MARK,R1(MARK,JJ-1),Z(MARK,JJ-1),DZ,TR,VR,CONST,LAMDA,
     *TE,I,J,JJ,ZATOM)
      DTR=TR*1.0D0
      CALL DQG32(0.D0,DTR,FETA2,Y112)
      VSUM(MARK)=VSUM(MARK)+VZ
      THETA(MARK,JJ)=THETA(MARK,JJ-1)+Y112*ANGMOM
      VR=-SQRT(VR*VR+ABS(CONST)*(1/R1(MARK,JJ)**2-1/R1(MARK,JJ-1)**2))
      VT=ANGMOM/R1(MARK,JJ)
      VX=VR*SIN(THETA(MARK,JJ))+VT*COS(THETA(MARK,JJ))
      VY=VR*COS(THETA(MARK,JJ))-VT*SIN(THETA(MARK,JJ))
      X=R1(MARK,JJ)*SIN(THETA(MARK,JJ))
      Y=R1(MARK,JJ)*COS(THETA(MARK,JJ))
      I=I-1
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      GOTO 4
C ---------------------------------------------------------------------
C        CASE 4 :N=NO(1-A**2/R**2), CONST>0.0, VR<0.0, FIND MIN. RADIUS
C        N=NO(1+A**2/R**2)
C ---------------------------------------------------------------------

  106 R1(MARK,JJ)=RMIN
      TR=TIMEPN(R1(MARK,JJ-1),R1(MARK,JJ),VR,CONST)

      VZ=SQRT((RI(I,J)*3E10)**2-VR*VR-(ANGMOM/R1(MARK,JJ-1))**2)
      Z(MARK,JJ)=VZ*TR+Z(MARK,JJ-1)
      IF (Z(MARK,JJ).GT.J*DZ) GOTO 102
      CALL ABOB2P(MARK,R1(MARK,JJ-1),Z(MARK,JJ-1),DZ,TR,VR,CONST,LAMDA,
     *TE,I,J,JJ,ZATOM)
      DTR=TR*1.0D0
      CALL DQG32(0.0D0,DTR,FETA2,Y112)
      VSUM(MARK)=VSUM(MARK)+VZ
      THETA(MARK,JJ)=THETA(MARK,JJ-1)+Y112*ANGMOM
      VR=0.0
      VT=ANGMOM/R1(MARK,JJ)
      VX=VR*SIN(THETA(MARK,JJ))+VT*COS(THETA(MARK,JJ))
      VY=VR*COS(THETA(MARK,JJ))-VT*SIN(THETA(MARK,JJ))
      X=R1(MARK,JJ)*SIN(THETA(MARK,JJ))
      Y=R1(MARK,JJ)*COS(THETA(MARK,JJ))
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      GOTO 3
  120 WRITE(6,612)
  612 FORMAT(/'*****RAY REACHES THE LOWEST LIMIT AND ROTATES AROUND THE
     *AXIS*****'/)
      R1(MARK,JJ)=RMIN
      TR=TIMENN(R1(MARK,JJ-1),R1(MARK,JJ),VR,CONST)
      VZ=SQRT((RI(I,J)*3E10)**2-VR*VR-(ANGMOM/R1(MARK,JJ-1))**2)
      Z(MARK,JJ)=VZ*TR+Z(MARK,JJ-1)
      IF (Z(MARK,JJ).GT.J*DZ) GOTO 1091
      CALL ABOB2N(MARK,R1(MARK,JJ-1),Z(MARK,JJ-1),DZ,TR,VR,CONST,LAMDA,
     *TE,I,J,JJ,ZATOM)
```

```
      VSUM(MARK)=VSUM(MARK)+VZ
      THETA(MARK,JJ)=THETA(MARK,JJ-1)+ANGMOM*TR/R1(MARK,JJ)**2
      GOTO 666
C-----------------------------------------------------------------------
C           ALGORITHM FOR FINDING RAY LOCATIONS WHEN IT REACHES THE INNER
C           MOST CORE SHELL
C-----------------------------------------------------------------------
    9 JJ=JJ+1
      IF (JJ.GT.100)GOTO 37
      VDOTR=X*VX+Y*VY
      TR1=DSQRT(VDOTR*VDOTR+(R2(1,J)*R2(1,J)-R1(MARK,JJ-1)**2)*(VX*VX+VY
     **VY))
      TR1=-VDOTR-SIGN(TR1,VDOTR)
      TR1=TR1/(VX*VX+VY*VY)
      TR=-2*VDOTR/(VX**2+VY**2)
      VZ=DSQRT(RI(I,J)*RI(I,J)*9E20-VX*VX-VY*VY)
      Z(MARK,JJ)=Z(MARK,JJ-1)+VZ*TR/2
      IF (Z(MARK,JJ).GT.J*DZ) GOTO 94
      GOTO 96
C -----------------------------------------------------------------------
C           RAY HITS VERTICAL BOUNDARY BEFORE IT REACHES THE MINIMUM
C -----------------------------------------------------------------------
   94 Z(MARK,JJ)=J*DZ
      TZ=(Z(MARK,JJ)-Z(MARK,JJ-1))/VZ
      VSUM(MARK)=VSUM(MARK)+VZ
      X=X+VX*TZ
      Y=Y+VY*TZ
      R1(MARK,J[illegible]X+Y*Y)
      THETA(MAR[illegible]X,Y)
      A1=2.19E3*TE(1,J)*SQRT(TE(1,J))*LAMDA
      A2=1.14E4*LAMDA*TE(1,J)
      ALAMDA=ALOG(AMIN1(A1,A2))
      A3=8.67E-30*LAMDA*LAMDA*ALAMDA/TE(1,J)
      KA(MARK,JJ)=A3*NOI(1,J)**2*3E10*TZ/SQRT(1-NOI(1,J)/CRIDEN)
      J=J+1
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      IF (J.GT.NX) GOTO 666
      GOTO 9
C-----------------------------------------------------------------------
C           TO CALCULATE THE MINIMUM POINT IN R
C-----------------------------------------------------------------------
   96 X=X+VX*TR/2
      Y=Y+VY*TR/2
      R1(MARK,JJ)=DSQRT(X**2+Y**2)
      THETA(MARK,JJ)=ANGLE(X,Y)
      A1=2.19E3*TE(1,J)*SQRT(TE(1,J))*LAMDA
      A2=1.14E4*LAMDA*TE(1,J)
      ALAMDA=ALOG(A[illegible]1,A2))
      A3=8.67E-30*LAMDA*LAMDA*ALAMDA/TE(1,J)
      KA(MARK,JJ)=A3*NOI(1,J)**2*3E10*(TR/2)/SQRT(1-NOI(1,J)/CRIDEN)
      TR2=TR
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      VSUM(MARK)=VSUM(MARK)+VZ
C -----------------------------------------------------------------------
C           TO LOCATE THE NEXT POINT AFTER HAVING REACHED THE MINIMUM
C           POINT
C -----------------------------------------------------------------------
  994 JJ=JJ+1
      IF (JJ.GT.100) GOTO 37
      Z(MARK,JJ)=Z(MARK,JJ-1)+VZ*TR2/2
   99 IF (Z(MARK,JJ).GT.J*DZ) GOTO 97
      GOTO 98
C -----------------------------------------------------------------------
C           RAY HITS VERTICAL BOUNDARY BEFORE IT REACHES THE SHELL SURFACE
C -----------------------------------------------------------------------
   97 Z(MARK,JJ)=J*DZ
      TZ=(Z(MARK,JJ)-Z(MARK,JJ-1))/VZ
      X=X+VX*TZ
```

```
      Y=Y+VY*TZ
      VSUM(MARK)=VSUM(MARK)+VZ
      R1(MARK,JJ)=DSQRT(X*X+Y*Y)
      THETA(MARK,JJ)=ANGLE(X,Y)
      A1=2.19E3*TE(1,J)*SQRT(TE(1,J))*LAMDA
      A2=1.14E4*LAMDA*TE(1,J)
      ALAMDA=ALOG(AMIN1(A1,A2))
      A3=8.67E-30*LAMDA*LAMDA*ALAMDA/TE(1,J)
      KA(MARK,JJ)=A3*NOI(1,J)**2*3E10*TZ/SQRT(1-NOI(1,J)/CRIDEN)
      J=J+1
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      IF (J.GT.NX) GOTO [illegible]
      TR2=TR2-2*TZ
      GOTO 994
C ---------------------------------------------------------------------
C        TO FIND THE POINT SYMMETRICAL TO THE FIRST ENTRY POINT AT THE
C        INNER MOST CORE (ITS RADIUS MIGHT BE SMALLER THAN THAT OF THE
C        1ST SHELL)
C ---------------------------------------------------------------------
   98 X=X+VX*TR2/2
      Y=Y+VY*TR2/2
      VSUM(MARK)=VSUM(MARK)+VZ
      R1(MARK,JJ)=DSQRT(X*X+Y*Y)
      THETA(MARK,JJ)=ANGLE(X,Y)
      A1=2.19E3*TE(1,J)*SQRT(TE(1,J))*LAMDA
      A2=1.14E4*LAMDA*TE(1,J)
      ALAMDA=ALOG(AMIN1(A1,A2))
      A3=8.67E-30*LAMDA*LAMDA*ALAMDA/TE(1,J)
      KA(MARK,JJ)=A3*NOI(1,J)**2*3E10*(TR2/2)/SQRT(1-NOI(1,J)/CRIDEN)
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      IF (ABS(R1(MARK,JJ)-R1(1,J)).GT.1.0E-4) GOTO 92
      I=2
      GOTO 3
C ---------------------------------------------------------------------
C        THE SYMMETRICAL POINT IS NOT RIGHT AT THE 1ST SHELL LAYER BUT
C        WITHIN IT
C ---------------------------------------------------------------------
   92 TZZ=0
  995 JJ=JJ+1
      IF (JJ.GT.100) GOTO 37
      Z(MARK,JJ)=Z(MARK,JJ-1)+VZ*(TR1-TR)
      IF (Z(MARK,JJ).GT.J*DZ) GOTO 991
      GOTO 992
C ---------------------------------------------------------------------
C        RAY HITS THE COLUMN BEFORE IT REACHES THE SYMMETRICAL POINT
C ---------------------------------------------------------------------
  991 Z(MARK,JJ)=J*DZ
      TZZ=(Z(MARK,JJ)-Z(MARK,JJ-1))/VZ
      X=X+VX*TZZ
      Y=Y+VY*TZZ
      VSUM(MARK)=VSUM(MARK)+VZ
      R1(MARK,JJ)=DSQRT(X*X+Y*Y)
      THETA(MARK,JJ)=ANGLE(X,Y)
      A1=2.19E3*TE(1,J)*SQRT(TE(1,J))*LAMDA
      A2=1.14E4*LAMDA*TE(1,J)
      ALAMDA=ALOG(AMIN1(A1,A2))
      A3=8.67E-30*LAMDA*LAMDA*ALAMDA/TE(1,J)
      KA(MARK,JJ)=A3*NOI(1,J)**2*3E10*TZZ/SQRT(1-NOI(1,J)/CRIDEN)
      J=J+1
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      IF (J.GT.NX) GOTO 666
      TR=TR+TZZ
      GOTO 995
C ---------------------------------------------------------------------
C        RAY HITS AT THE 1ST SHELL LAYER
C ---------------------------------------------------------------------
```

```
  992 X=X+VX*(TR1-TR)
      Y=Y+VY*(TR1-TR)
      VSUM(MARK)=VSUM(MARK)+VZ
      R1(MARK,JJ)=R2(1,J)
      THETA(MARK,JJ)=ANGLE(X,Y)
  999 A1=2.19E3*TE(1,J)*SQRT(TE(1,J))*LAMDA
      A2=1.14E4*LAMDA*TE(1,J)
      ALAMDA=ALOG(AMIN1(A1,A2))
      A3=8.67E-30*LAMDA*LAMDA*ALAMDA/TE(1,J)
      KA(MARK,JJ)=A3*NOI(1,J)**2*3E10*(TR1-TR)/SQRT(1-NOI(1,J)/
     *CRIDEN)
      WRITE(6,605)MARK,J,KA(MARK,JJ)
  605 FORMAT(/'ABS.COEFF.(',I2,', 1,',I2,')=',E15.8)
      I=2
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      GOTO 3
  993 R1(MARK,JJ)=R2(1,J)
      TR=(R1(MARK,JJ)-R1(MARK,JJ-1))/(VR*RI(1,J))
      Z(MARK,JJ)=Z(MARK,JJ-1)+TR*VZ*RI(1,J)
      IF (Z(MARK,JJ).GT.J*DZ) GOTO 996
      VSUM(MARK)=VSUM(MARK)+VZ
      X=X+VX*TR*RI(1,J)
      Y=Y+VY*TR*RI(1,J)
      THETA(MARK,JJ)=ANGLE(X,Y)
      A1=2.19E3*TE(1,J)*SQRT(TE(1,J))*LAMDA
      A2=1.14E4*LAMDA*TE(1,J)
      ALAMDA=ALOG(AMIN1(A1,A2))
      A3=8.67E-30*LAMDA*LAMDA*ALAMDA/TE(1,J)
      KA(MARK,JJ)=A3*NOI(1,J)**2*3E10*TR/SQRT(1-NOI(1,J)/CRIDEN)
      WRITE(6,605)MARK,J,KA(MARK,JJ)
      I=2
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      GOTO 999
  996 Z(MARK,JJ)=J*DZ
      TZZ=(Z(MARK,JJ)-Z(MARK,JJ-1))/(VZ*RI(1,J))
      X=X+VX*RI(1,J)*TZZ
      Y=Y+VY*RI(1,J)*TZZ
      VSUM(MARK)=VSUM(MARK)+VZ
      R1(MARK,JJ)=DSQRT(X*X+Y*Y)
      THETA(MARK,JJ)=ANGLE(X,Y)
      A1=2.19E3*TE(1,J)*SQRT(TE(1,J))*LAMDA
      A2=1.14E4*LAMDA*TE(1,J)
      ALAMDA=ALOG(AMIN1(A1,A2))
      A3=8.67E-30*LAMDA*LAMDA*ALAMDA/TE(1,J)
      KA(MARK,JJ)=A3*NOI(1,J)**2*3E10*TZZ/SQRT(1-NOI(1,J)/CRIDEN)
      J=J+1
      LOCX(MARK,JJ)=I
      LOCY(MARK,JJ)=J
      IF (J.GT.NX) GOTO 666
      JJ=JJ+1
      IF (JJ.GT.100) GOTO 37
      GOTO 993
C-----------------------------------------------------------------------
C          ALGORITHM FOR RAYS GOING ALONG Z-AXIS
C-----------------------------------------------------------------------
  110 JJ=1
      J=1
      VZ=3.0E10
      DZ=ZL/NX1
      R1(MARK,1)=0.0
      THETA(MARK,1)=0.0
      Z(MARK,1)=0.0
      LOCX(MARK,1)=1
      LOCY(MARK,1)=1
 1101 JJ=JJ+1
      IF (JJ.GT.100) GOTO 37
      TZ=DZ/(VZ*RI(1,J))
      R1(MARK,JJ)=0.0
      THETA(MARK,JJ)=0.0
```

```
      Z(MARK,JJ)=Z(MARK,JJ-1)+DZ
      VSUM(MARK)=VSUM(MARK)+VZ
      A1=2.19E3*TE(1,J)*SQRT(TE(1,J))*LAMDA
      A2=1.14E4*LAMDA*TE(1,J)
      ALAMDA=ALOG(AMIN1(A1,A2))
      A3=8.67E-30*LAMDA*LAMDA*ALAMDA/TE(1,J)
      KA(MARK,JJ)=A3*NOI(1,J)**2*3E10*TZ/SQRT(1-NOI(1,J)/CRIDEN)
      J=J+1
      LOCX(MARK,JJ)=1
      LOCY(MARK,JJ)=J
      IF (J.GT.NX) GOTO 666
      GOTO 1101
  666 NPTS(MARK)=JJ
      WRITE(6,601)MARK
  601 FORMAT(/'DATA FROM PROGRAM:RAYABS',/'RAY',1X,I3/' RAD.POS.  ',2X,
     *'ANG.POS.  ',2X,
     *'AXIAL POSI.',2X,'ABS.COEF.  ',2X,'X CELL LOC.',2X,'Y CELL LOC.')
      WRITE(6,602) (R1(MARK,J),THETA(MARK,J),Z(MARK,J),KA(MARK,J),LOCX(M
     *ARK,J),LOCY(MARK,J),J=1,JJ)
  602 FORMAT(E10.3,2X,E10.3,2X,E10.3,2X,E10.3,2X,7X,I3,2X,7X,I3)
      WRITE(2,603) (R1(MARK,J),Z(MARK,J),J=1,JJ)
  603 FORMAT(2E18.10)
      VSUM(MARK)=VSUM(MARK)/JJ
      GOTO 12
      END
C--------------------------------------------------------------------
C           X VELOCITY ALGORITHM
C--------------------------------------------------------------------
      REAL FUNCTION XVEL(X,VX,T,I,J)
      DOUBLE PRECISION X
      COMMON /GRIDP/R2(60,60),OMEGA(60,60),LASHEL
      XVEL=-OMEGA(I,J)*X*SIN(OMEGA(I,J)*T)+VX*COS(OMEGA(I,J)*T)
      RETURN
      END
C--------------------------------------------------------------------
C           Y VELOCITY ALGORITHM
C--------------------------------------------------------------------
      REAL FUNCTION YVEL(Y,VY,T,I,J)
      DOUBLE PRECISION Y
      COMMON /GRIDP/R2(60,60),OMEGA(60,60),LASHEL
      YVEL=-OMEGA(I,J)*Y*SIN(OMEGA(I,J)*T)+VY*COS(OMEGA(I,J)*T)
      RETURN
      END
C--------------------------------------------------------------------
C           X COORDINATE ALGORITHM
C--------------------------------------------------------------------
      REAL FUNCTION XCOOR(X,VX,T,I,J)
      DOUBLE PRECISION X
      COMMON /GRIDP/R2(60,60),OMEGA(60,60),LASHEL
      XCOOR=X*COS(OMEGA(I,J)*T)+VX*SIN(OMEGA(I,J)*T)/OMEGA(I,J)
      RETURN
      END
C--------------------------------------------------------------------
C           Y COORDINATE ALGORITHM
C--------------------------------------------------------------------
      REAL FUNCTION YCOOR(Y,VY,T,I,J)
      DOUBLE PRECISION Y
      COMMON /GRIDP/R2(60,60),OMEGA(60,60),LASHEL
      YCOOR=Y*COS(OMEGA(I,J)*T)+VY*SIN(OMEGA(I,J)*T)/OMEGA(I,J)
      RETURN
      END
C--------------------------------------------------------------------
C           MINIMUM RADIUS ALGORITHM
C--------------------------------------------------------------------
      REAL FUNCTION SMALR(R,VX,VY,D,I,J)
      DOUBLE PRECISION VX,VY
      COMMON /GRIDP/R2(60,60),OMEGA(60,60),LASHEL
      SMALR=R**2/2+(VX**2+VY**2)/(2*OMEGA(I,J)**2)-D
      IF (SMALR.LT.0.0) SMALR=0.0
      SMALR=SQRT(SMALR)
      RETURN
      END
```

```
C-----------------------------------------------------------------------
C          MAXIMUM RADIUS ALGORITHM
C-----------------------------------------------------------------------
      REAL FUNCTION BIGR(R,VX,VY,D,I,J)
      DOUBLE PRECISION VX,VY
      COMMON /GRIDP/R2(60,60),OMEGA(60,60),LASHEL
      BIGR=DSQRT(R**2/2+(VX**2+VY**2)/(2*OMEGA(I,J)**2)+D)
      RETURN
      END
C-----------------------------------------------------------------------
C          TIME FOR TRAVERSING FROM SHELL TO SHELL
C-----------------------------------------------------------------------
      REAL FUNCTION RTIME(RNOT,RNEW,VELX,VELY,ID,JD,PHIO,DD)
      DOUBLE PRECISION VELX,VELY
      INTEGER FLAG
      COMMON /GRIDP/R2(60,60),OMEGA(60,60),LASHEL
      IO=ID
      RTIME1=RNEW**2-RNOT**2/2-(VELX**2+VELY**2)/(2*OMEGA(ID,JD)**2)
      RTIME1=ASIN(RTIME1/DD)
      THETA=RTIME1
      FLAG=1
      IO=0
      IF (THETA.LT.0.0) GOTO 1
C ----------------------------------------------------------------------
C          TO CONVERT THETA TO ITS MULTIPLE ANGLE IN THE 2ND QUADRANT
C          GIVEN THE ARGUMENT OF SIN(THETA) IS POSITIVE
C ----------------------------------------------------------------------
    3 RTIME=(RTIME1-PHIO)/(2*OMEGA(ID,JD))
      IF (RTIME.GT.0.0) RETURN
      IO=IO+1
      IF (FLAG.EQ.0)GOTO 2
      RTIME1=IO*3.14159-THETA
      FLAG=0
      GOTO 3
C ----------------------------------------------------------------------
C          TO CONVERT THETA TO ITS MULTIPLE VALUE IN THE 3RD QUADRANT
C          GIVEN THE ARGUMENT OF SIN(THETA) IS POSITIVE
C ----------------------------------------------------------------------
    2 RTIME1=IO*3.14159+THETA
      FLAG=1
      GOTO 3
C ----------------------------------------------------------------------
C          TO CONVERT THETA TO ITS MULTIPLE VALUE IN THE 4TH QUADRANT
C          GIVEN THE ARGUMENT OF SIN(THETA) IS NEGATIVE
C-----------------------------------------------------------------------
    1 IO=IO+1
      RTIME1=3.14159+ABS(THETA)
    5 RTIME=(RTIME1-PHIO)/(2*OMEGA(ID,JD))
      IF (RTIME.GT.0.0) RETURN
      IO=IO+1
      IF (FLAG.EQ.0) GOTO 4
C ----------------------------------------------------------------------
C          TO CONVERT THETA TO ITS MULTIPLE VALUE IN THE 2ND QUADRANT
C          GIVEN THE ARGUMENT OF SIN(THETA) IS NEGATIVE
C ----------------------------------------------------------------------
      RTIME1=IO*3.14159-ABS(THETA)
      FLAG=0
      GOTO 5
C ----------------------------------------------------------------------
C          TO CONVERT THETA TO ITS MULTIPLE VALUE IN THE 3RD QUADRANT
C          GIVEN THE ARGUMENT OF SIN(THETA) IS NEGATIVE
C ----------------------------------------------------------------------
    4 RTIME1=IO*3.14159+ABS(THETA)
      FLAG=1
      GOTO 5
      END
C-----------------------------------------------------------------------
C          ANGLE BETWEEN TWO CONSECUTIVE POINTS
C-----------------------------------------------------------------------
      REAL FUNCTION ANGLE(X,Y)
      DOUBLE PRECISION X,Y
```

```
      IF (X.EQ.0.0.AND.Y.GE.0.0) GOTO 2
      IF (X.EQ.0.0.AND.Y.LT.0.0) GOTO 6
      IF (Y.EQ.0.0.AND.X.GT.0.0) GOTO 4
      IF (Y.EQ.0.0.AND.X.LT.0.0) GOTO 7
      ANGLE=DATAN(X/Y)
      GOTO 5
    2 ANGLE=0.0
      RETURN
    6 ANGLE=3.14159
      RETURN
    4 ANGLE=1.570795
      RETURN
    7 ANGLE=4.712385
      RETURN
    5 IF (ANGLE.LT.0.0) GOTO 1
      IF (X.GT.0.0.AND.Y.GT.0.0) RETURN
C ----------------------------------------------------------------------
C           ANGLE IN THE 3RD QUADRANT WITH ARCTAN(X/Y) BEING POSITIVE
C ----------------------------------------------------------------------
      ANGLE=ANGLE+3.14159
      RETURN
    1 IF (X.LT.0.0.AND.Y.GT.0.0) GOTO 3
C ----------------------------------------------------------------------
C           ANGLE IN THE 2ND QUADRANT WITH X>0,Y<0
C ----------------------------------------------------------------------
      ANGLE=3.14159-ABS(ANGLE)
      RETURN
C ----------------------------------------------------------------------
C           ANGLE IN THE 4TH QUADRANT WITH X<0,Y>0
C ----------------------------------------------------------------------
    3 ANGLE=2*3.14159-ABS(ANGLE)
      RETURN
      END
C-----------------------------------------------------------------------
C                     ALGORITHM FOR PHASE ANGLE PHI
C-----------------------------------------------------------------------
      REAL FUNCTION FI(T1,T2)
      [illegible]EQ.0.0) GOTO 4
      [illegible]T2)
      G[illegible]
    4 FI=1.[illegible]795
    3 IF (FI.GT.0.0) GOTO 1
      IF (T1.GT.0.0.AND.T2.LT.0.0) GOTO 2
C ----------------------------------------------------------------------
C           PHI HAS THE MULTIPLE VALUE IN THE 4TH QUADRANT FOR PHI BEING
C           NEGATIVE
C ----------------------------------------------------------------------
      FI=2*3.14159-ABS(FI)
      RETURN
C ----------------------------------------------------------------------
C           PHI HAS THE MULTIPLE VALUE IN THE 2ND QUADRANT FOR PHI BEING
C           NEGATIVE
C ----------------------------------------------------------------------
    2 FI=3.14159-ABS(FI)
      RETURN
C ----------------------------------------------------------------------
C           PHI HAS THE MULTIPLE VALUEIN THE 3RD QUADRANT FOR PHI BEING
C           POSITIVE
C ----------------------------------------------------------------------
    1 IF (T1.LT.0.0.AND.T2.LT.0.0) FI=3.14159+ABS(FI)
      RETURN
      END
C-----------------------------------------------------------------------
C           ALGORITHM FOR FINDING THE TIME OF TRAVERSING SHELL
C           WITH A DECREASING DENSITY  N=NO(1-R**2/A**2)
C-----------------------------------------------------------------------
      REAL FUNCTION HTIME(A,B,C,I,J,NC)
      COMMON /GRIDP/R2(60,60),OMEGA(60,60),LASHEL
      CALL ROOT(A,B,C,R1,R22,NC)
      IF (NC.EQ.1) RETURN
      IF (R1.LT.0.0.AND.R22.LT.0.0) GOTO 2
```

```
C-----------------------------------------------------------------------
C         TIME FUNCTION FOR NON-PARABOLIC DENSTIY APPROXIMATION
C         N=NO(1-A**2/R**2)
C
C         CONST>0.0, VR>0.0
C -----------------------------------------------------------------------
      FUNCTION TIMEPP(RO,R1,VR,CONST)
      FACT=(RO*VR)**2+CONST
      TIMEPP=FACT*R1**2-CONST*RO**2
      TIMEPP=RO/FACT*SQRT(TIMEPP)-RO*RO*RO*VR/FACT
      RETURN
      END
C-----------------------------------------------------------------------
C         CONST>0.0, VR<0.0
C -----------------------------------------------------------------------

      FUNCTION TIMEPN(RO,R1,VR,CONST)
      FACT=(RO*VR)**2+CONST
      TIMEPN=FACT*R1**2-CONST*RO**2
      IF (TIMEPN.LT.0.0) TIMEPN=0.0
      TIMEPN=RO/FACT*(-SQRT(TIMEPN))+RO*RO*RO*ABS(VR)/FACT
      RETURN
      END
C -----------------------------------------------------------------------
C         CONST<0.0, VR>0.0
C -----------------------------------------------------------------------

      FUNCTION TIMENP(RO,R1,VR,CONST)
      FACT=(RO*VR)**2-ABS(CONST)
      TIMENP=FACT*R1**2+ABS(CONST)*RO**2
      TIMENP=RO/FACT*SQRT(TIMENP)-RO*RO*RO*VR/FACT
      RETURN
      END
C -----------------------------------------------------------------------
C         CONST<0.0, VR<0.0,
C -----------------------------------------------------------------------
      FUNCTION TIMENN(RO,R1,VR,CONST)
      FACT=(RO*VR)**2-ABS(CONST)
      TIMENN=FACT*R1**2+ABS(CONST)*RO**2
      TIMENN=RO/FACT*(-SQRT(TIMENN))+RO*RO*RO*ABS(VR)/FACT
      RETURN
      END
C-----------------------------------------------------------------------
C         ALGORITHM FOR FINDING THE ABSORPTION COEFFICIENTS
C         IN THE PLASMA CELLS WITH A RADIALLY INCREASING PARABOLIC
C         DENSITY PROFILE N=NO(1+R**2/A**2)
C-----------------------------------------------------------------------
      SUBROUTINE ABSORB(MRAY,RO,ZO,DZ,T1,VX,VY,PHI,LAMDA,TE,D,I,J,JJ,Z)
      DOUBLE PRECISION DRO,DOMEGA,DAOISO,DNOI,DD,F,Y,VX,T2,VY,
     *DVX,DVY
      REAL NOI,LAMDA,A1,A2,LLAMDA,KA,TE(30,60)
      INTEGER TOTRAY,MRAY,Z
      EXTERNAL F
      COMMON /ABSOB/AOISO(60,60),NOI(60,60),LOCX(100,100),LOCY(100,100),
     *KA(100,100)
      COMMON /FUNCF/DRO,DVX,DVY,DOMEGA,DAOISO,DNOI,DD,FI
      COMMON /GRIDP/R2(60,60),OMEGA(60,60),LASHEL
C -----------------------------------------------------------------------
C         CHANGE VALUES INTO DOUBLE PRECISION VALUES
C -----------------------------------------------------------------------
      DRO=RO*1.DO
C     DR1=R1*1.DO
      DAOISO=AOISO(I,J)*1.DO
      DNOI=NOI(I,J)*1.DO
C     DVZ=VZ*1.DO
C     DVT=(VX**2+VY**2)*1.DO
      DD=D*1.DO
      DOMEGA=OMEGA(I,J)*1.DO
      T2=T1*1.ODO
```

```
      DVX=VX
      DVY=VY
      FI=PHI
C -----------------------------------------------------------------------
C       FIND THE INTEGRAL FORM OF ABSORPTION BETWEEN TWO POINTS
C       LOCX(MRAY,JJ) DENOTES THE CORRESPONDING CELL IN A PLASMA SHELL
C       LAYER
C -----------------------------------------------------------------------
      CALL DQG32(O.DO,T2,F,Y)
      IF (LOCX(MRAY,JJ-1).LT.I) GOTO 3
      IX=INT((LOCX(MRAY,JJ-1)-1)/2.0)+1
      GOTO 4
    3 IX=INT(LOCX(MRAY,JJ-1)/2.0)+1
    4 TE15=TE(IX,J)*SQRT(TE(IX,J))
      A1=2.19E3*TE15*LAMDA
      A2=1.14E4*LAMDA*TE(IX,J)
      LLAMDA=ALOG(AMIN1(A1,A2))
      A3=26.01E-20*LAMDA**2*LLAMDA*Z
      TKA=DABS(Y)*A3/TE15
C -----------------------------------------------------------------------
C       ABSORPTION COEF. IS THE SUM OF ALL SEGMENT LENGTHS WITHIN A
C       PLASMA CELL
C -----------------------------------------------------------------------
      KA(MRAY,JJ)=TKA
      RETURN
      END

C-----------------------------------------------------------------------
C       ALGORITHM  FOR THE FUNCTION USED IN THE ABSORPTION
C       INTEGRAL  N=NO(1+R**2/A**2)
C-----------------------------------------------------------------------
      DOUBLE PRECISION FUNCTION F(T)
      DOUBLE PRECISION T,RSQ,DRO,VX,VY,DOMEGA,DNOI,DAOISQ,DD
      COMMON /FUNCF/DRO,VX,VY,DOMEGA,DAOISQ,DNOI,DD,PHI
      COMMON /DNGRP/RI(60,60),CRIDEN,FLAG(60,60)
      RSQ=DRO**2/2.ODO+(VX**2+VY**2)/(2*DOMEGA**2)+DD*1.ODO*DSIN(2*DOMEG
     *A*T+PHI)
      F=DNOI*(1.ODO+RSQ/DAOISQ)
      F=F*F
      RETURN
      END
C-----------------------------------------------------------------------
C       ALGORITHM FOR FINDING ABSORPTION COEF. FOR A
C       PARABOLIC DENSITY PROFILE  N=NO(1-R**2/A**2)
C-----------------------------------------------------------------------
      SUBROUTINE ABSOB1(MARK,T1,VX,VY,VZ,LAMDA,TE,TERM1,TERM2,
     *I,J,JJ,Z,ZO,DZ)
      REAL NOI,LAMDA,A1,A2,LLAMDA,KA,TE(30,60)
      INTEGER Z
      DOUBLE PRECISION DOMEGA,DAOISQ,DNOI,T2,G,Y,VX,VY,VELX,VELY
      EXTERNAL G
      COMMON /ABSOB/AOISQ(60,60),NOI(60,60),LOCX(100,100),LOCY(100,100),
     *KA(100,100)
      COMMON /FUNCG/VELX,VELY,DOMEGA,DAOISQ,DNOI,TEM1,TEM2
      COMMON /GRIDP/R2(60,60),OMEGA(60,60),LASHEL
      COMMON /DRAYP/XD(100),YD(100),THETAX(100),THETAY(100),P(100),TOTRA
     *Y,ENIN,P1,EN(100)
      VELX=VX
      VELY=VY
      TEM1=TERM1
      TEM2=TERM2
      DAOISQ=AOISQ(I,J)*1.DO
      DNOI=NOI(I,J)*1.DO
      DOMEGA=OMEGA(I,J)*1.DO
      T2=T1*1.ODO
      CALL DQG32(O.DO,T2,G,Y)
      IF (LOCX(MARK,JJ-1).LT.I) GOTO 3
      IX=IFIX((LOCX(MARK,JJ-1)-1)/2.0)+1
      GOTO 4
    3 IX=IFIX(LOCX(MARK,JJ-1)/2.0)+1
    4 TE15=TE(IX,J)*SQRT(TE(IX,J))
```

```
      A1=2.19E3*TE15*LAMDA
      A2=1.14E4*LAMDA*TE(IX,J)
      LLAMDA=ALOG(AMIN1(A1,A2))
      A3=26.01E-20*LAMDA**2*LLAMDA*Z
      TKA=DABS(Y)*A3/TE15
      KA(MARK,JJ)=TKA
      RETURN
      END
C -----------------------------------------------------------------------
C          FUNCTION FOR NON-PARABOLIC DE/IN-CREASING DENSITY
C          N=NO(1-A**2/R**2),CONST>0.0
C -----------------------------------------------------------------------
      SUBROUTINE ABOB2P(MRAY,RO,ZO,DZ,T1,VR,CONST,LAMDA,TE,I,J,JJ,Z)
      DOUBLE PRECISION DRO,DVR,DCONST,DNOI,DAOISQ,Y,F2P,T2,F2PN
      DOUBLE PRECISION F2PPN,F2PNN
      REAL NOI,LAMDA,TE(30,60),KA,LLAMDA
      INTEGER MRAY,Z,FLAG
      EXTERNAL F2P
      EXTERNAL F2PN
      EXTERNAL F2PPN
      EXTERNAL F2PNN
      COMMON /ABSOB/AOISQ(60,60),NOI(60,60),LOCX(100,100),LOCY(100,100),
     *KA(100,100)
      COMMON /FUNF2P/DRO,DNOI,DAOISQ,DCONST,DVR
      COMMON /GRIDP/R2(60,60),OMEGA(60,60),LASHEL
      COMMON /DNGRP/RI(60,60),CRIDEN,FLAG(60,60)
      DRO=RO*1.D0
      DVR=VR*1.D0
      DNOI=NOI(I,J)*1.D0
      DCONST=CONST*1.D0
      DAOISQ=AOISQ(I,J)*1.D0
      T2=T1*1.0D0
      IF (FLAG(I,J).EQ.4) GOTO 6
      IF (VR.LT.0.0) GOTO 5
      CALL DQG32(0.D0,T2,F2P,Y)
      GOTO 2
    5 CALL DQG32(0.D0,T2,F2PN,Y)
      GOTO 2
    6 IF (VR.LT.0.0) GOTO 7
      CALL DQG32(0.D0,T2,F2PPN,Y)
      GOTO 2
    7 CALL DQG32(0.D0,T2,F2PNN,Y)
    2 IF (LOCX(MRAY,JJ-1).LT.I) GOTO 3
      IX=INT((LOCX(MRAY,JJ-1)-1)/2.0)+1
      GOTO 4
    3 IX=INT(LOCX(MRAY,JJ-1)/2.0)+1
    4 TE15=TE(IX,J)*SQRT(TE(IX,J))
      A1=2.19E3*TE15*LAMDA
      A2=1.14E4*LAMDA*TE(IX,J)
      LLAMDA=ALOG(AMIN1(A1,A2))
      A3=26.01E-20*LAMDA**2*LLAMDA*Z
      TKA=DABS(Y)*A3/TE15
      KA(MRAY,JJ)=TKA
      RETURN
      END
C-----------------------------------------------------------------------
C          FUNCTION FOR NON-PARABOLIC DE/INCREASING DENSITY
C          N=NO(1-A**2/R**2) CONST<0.0
C-----------------------------------------------------------------------
      SUBROUTINE ABOB2N(MRAY,RO,ZO,DZ,T1,VR,CONST,LAMDA,TE,I,J,JJ,Z)
      DOUBLE PRECISION DRO,DVR,DCONST,DNOI,DAOISQ,Y,F2N,T2,F2P,F2PN
      REAL NOI,LAMDA,TE(30,60),KA,LLAMDA
      INTEGER MRAY,Z,FLAG
      EXTERNAL F2P
      EXTERNAL F2PN
      COMMON /ABSOB/AOISQ(60,60),NOI(60,60),LOCX(100,100),LOCY(100,100),
     *KA(100,100)
      COMMON /FUNF2P/DRO,DNOI,DAOISQ,DCONST,DVR
      COMMON /GRIDP/R2(60,60),OMEGA(60,60),LASHEL
      COMMON /DNGRP/RI(60,60),CRIDEN,FLAG(60,60)
```

```
      DRO=RO*1.DO
      DVR=VR*1.DO
      DNOI=NOI(I,J)*1.DO
      DCONST=CONST*1.DO
      DAOISQ=AOISQ(I,J)*1.DO
      T2=T1*1.ODO
      IF (VR.LT.O.O) GOTO 5
      CALL DQG32(O.DO,T2,F2P,Y)
      GOTO 2
    5 CALL DQG32(O.DO,T2,F2PN,Y)
    2 IF (LOCX(MRAY,JJ-1).LT.I) GOTO 3
      IX=INT((LOCX(MRAY,JJ-1)-1)/2.O)+1
      GOTO 4
    3 IX=INT(LOCX(MRAY,JJ-1)/2.O)+1
    4 TE15=TE(IX,J)*SQRT(TE(IX,J))
      A1=2.19E3*TE15*LAMDA
      A2=1.14E4*LAMDA*TE(IX,J)
      LLAMDA=ALOG(AMIN1(A1,A2))
      A3=26.01E-20*LAMDA**2*LLAMDA*Z
      TKA=DABS(Y)*A3/TE15
      KA(MRAY,JJ)=TKA
      RETURN
      END
C-----------------------------------------------------------------------
C           FUNCTION F2P FOR N=NO(1-A**2/R**2)
C           CONST>O.O OR <O.O  VR>O.O
C-----------------------------------------------------------------------
      DOUBLE PRECISION FUNCTION F2P(T)
      DOUBLE PRECISION DRO,DNOI,DAOISQ,DCONST,DVR,D1
      COMMON /DNGRP/RI(60,60),CRIDEN,FLAG(60,60)
      COMMON /FUNF2P/DRO,DNOI,DAOISQ,DCONST,DVR
      D1=DRO*DRO*DVR*DVR+DCONST
      F2P=(T+DRO*DRO*DRO*DVR/D1)**2*D1*D1/DRO**2
      F2P=F2P+DCONST*DRO**2
      F2P=F2P/D1
      F2P=DNOI*(1-DAOISQ/F2P)
      F2P=F2P*F2P
      RETURN
      END
C-----------------------------------------------------------------------
C                     FUNCTION FOR N=NO(1-A**2/R**2)
C                     CONST>O.O OR <O.O VR<O.O
C-----------------------------------------------------------------------
      DOUBLE PRECISION FUNCTION F2PN(T)
      DOUBLE PRECISION DRO,DNOI,DAOISQ,DARG,DCONST,DVR,D1
      COMMON /DNGRP/RI(60,60),CRIDEN,FLAG(60,60)
      COMMON /FUNF2P/DRO,DNOI,DAOISQ,DCONST,DVR
      D1=DRO*DRO*DVR*DVR+DCONST
      F2PN=(T-DRO*DRO*DRO*DVR/D1)**2*D1*D1/DRO**2
      F2PN=F2PN+DCONST*DRO**2
      F2PN=F2PN/D1
      F2PN=DNOI*(1-DAOISQ/F2PN)
      F2PN=F2PN*F2PN
      RETURN
      END
C-----------------------------------------------------------------------
C          FUNCTION FOR N=NO(1+A**2/R**2)
C          CONST>O.O VR<O.O
C-----------------------------------------------------------------------
      DOUBLE PRECISION FUNCTION F2PNN(T)
      DOUBLE PRECISION DRO,DNOI,DAOISQ,DARG,DCONST,DVR,D1
      COMMON /DNGRP/RI(60,60),CRIDEN,FLAG(60,60)
      COMMON /FUNF2P/DRO,DNOI,DAOISQ,DCONST,DVR
      D1=DRO*DRO*DVR*DVR+DCONST
      F2PNN=(T-DRO*DRO*DRO*DVR/D1)**2*D1*D1/DRO**2
      F2PNN=F2PNN+DCONST*DRO**2
      F2PNN=F2PNN/D1
      F2PNN=DNOI*(1+DAOISQ/F2PNN)
      F2PNN=F2PNN*F2PNN
      RETURN
      END
```

```
C-----------------------------------------------------------------------
C          FUNCTION F2PPN FOR N=NO(1+A**2/R**2)
C          CONST>0.0 VR>0.0
C-----------------------------------------------------------------------
      DOUBLE PRECISION FUNCTION F2PPN(T)
      DOUBLE PRECISION DRO,DNOI,DAOISQ,DARG,DCONST,DVR,D1
      COMMON /DNGRP/RI(60,60),CRIDEN,FLAG(60,60)
      COMMON /FUNF2P/DRO,DNOI,DAOISQ,DCONST,DVR
      D1=DRO*DRO*DVR*DVR+DCONST
      F2PPN=(T+DRO*DRO*DRO*DVR/D1)**2*D1*D1/DRO**2
      F2PPN=F2PPN+DCONST*DRO**2
      F2PPN=F2PPN/D1
      F2PPN=DNOI*(1+DAOISQ/F2PPN)
      F2PPN=F2PPN*F2PPN
      RETURN
      END
C-----------------------------------------------------------------------
C          ABSORPTION ALGORITHM FOR N=NO(1-R**2/A**2)
C-----------------------------------------------------------------------
      DOUBLE PRECISION FUNCTION G(T)
      DOUBLE PRECISION T,ARG,RSQ,VX,VY,DOMEGA,DNOI,DAOISQ
      COMMON /DNGRP/RI(60,60),CRIDEN,FLAG(60,60)
      COMMON /FUNCG/VX,VY,DOMEGA,DAOISQ,DNOI,TERM1,TERM2
      ARG=DSINH(2*DOMEGA*T)
      RSQ=TERM1*(1+DSQRT(1+ARG**2))+TERM2*ARG-(VX**2+VY**2)/DOMEGA**2
      G=DNOI*(1.DO-RSQ/DAOISQ)
      G=G*G
      RETURN
      END
C-----------------------------------------------------------------------
C          FUNCTION FETA1 FOR N=NO(1-A**2/R**2)
C          CONST>0.0 OR <0.0  VR>0.0
C-----------------------------------------------------------------------
      DOUBLE PRECISION FUNCTION FETA1(T)
      DOUBLE PRECISION DRO,DNOI,DAOISQ,DARG,DCONST,DVR,D1
      COMMON /DNGRP/RI(60,60),CRIDEN,FLAG(60,60)
      COMMON /FUNF2P/DRO,DNOI,DAOISQ,DCONST,DVR
      D1=DRO*DRO*DVR*DVR+DCONST
      FETA1=(T+DRO*DRO*DRO*DVR/D1)**2*D1*D1/DRO**2
      FETA1=FETA1+DCONST*DRO**2
      FETA1=D1/FETA1
      RETURN
      END
C-----------------------------------------------------------------------
C          FUNCTION FOR N=NO(1-A**2/R**2)
C          CONST>0.0 OR <0.0 VR<0.0
C-----------------------------------------------------------------------
      DOUBLE PRECISION FUNCTION FETA3(T)
      DOUBLE PRECISION DRO,DNOI,DAOISQ,DARG,DCONST,DVR,D1
      COMMON /DNGRP/RI(60,60),CRIDEN,FLAG(60,60)
      COMMON /FUNF2P/DRO,DNOI,DAOISQ,DCONST,DVR
      D1=DRO*DRO*DVR*DVR+DCONST
      FETA3=(T-DRO*DRO*DRO*DVR/D1)**2*D1*D1/DRO**2
      FETA3=FETA3+DCONST*DRO**2
      FETA3=D1/FETA3
      RETURN
      END
C-----------------------------------------------------------------------
C          FUNCTION FOR N=NO(1+A**2/R**2)
C          CONST>0.0 VR<0.0
C-----------------------------------------------------------------------
      DOUBLE PRECISION FUNCTION FETA4(T)
      DOUBLE PRECISION DRO,DNOI,DAOISQ,DARG,DCONST,DVR,D1
      COMMON /DNGRP/RI(60,60),CRIDEN,FLAG(60,60)
      COMMON /FUNF2P/DRO,DNOI,DAOISQ,DCONST,DVR
      D1=DRO*DRO*DVR*DVR+DCONST
      FETA4=(T-DRO*DRO*DRO*DVR/D1)**2*D1*D1/DRO**2
      FETA4=FETA4+DCONST*DRO**2
      FETA4=D1/FETA4
      RETURN
      END
```

```
C-----------------------------------------------------------------------
C         FUNCTION FETA2 FOR N=NO(1+A**2/R**2)
C         CONST>0.0 VR>0.0
C-----------------------------------------------------------------------
      DOUBLE PRECISION FUNCTION FETA2(T)
      DOUBLE PRECISION DRO,DNOI,DAOISQ,DARG,DCONST,DVR,D1
      COMMON /DNGRP/RI(60,60),CRIDEN,FLAG(60,60)
      COMMON /FUNF2P/DRO,DNOI,DAOISQ,DCONST,DVR
      D1=DRO*DRO*DVR*DVR+DCONST
      FETA2=(T+DRO*DRO*DRO*DVR/D1)**2*D1*D1/DRO**2
      FETA2=FETA2+DCONST*DRO**2
      FETA2=D1/FETA2
      RETURN
      END
```