NAME OF AUTHOR/*NOM DE L'AUTEUR* RICHARD DOUGLAS MILROY

TITLE OF THESIS/*TITRE DE LA THÈSE* NUMERICAL AND ANALYTIC STUDIES OF LASER - PLASMA INTERACTIONS IN A SOLENOIDAL MAGNETIC FIELD

UNIVERSITY/*UNIVERSITÉ* UNIVERSITY OF ALBERTA

DEGREE FOR WHICH THESIS WAS PRESENTED/ *GRADE POUR LEQUEL CETTE THÈSE FUT PRÉSENTÉE* Ph. D

YEAR THIS DEGREE CONFERRED/*ANNÉE D'OBTENTION DE CE GRADE* 1978

NAME OF SUPERVISOR/*NOM DU DIRECTEUR DE THÈSE* DR. C.E. CAPJACK & Dr C.R. James

DATED/*DATÉ* SEPTEMBER 11, 1978 SIGNED/*SIGNÉ* Richard D. Milroy

PERMANENT ADDRESS/*RÉSIDENCE FIXE* 2735 - 43 ST S.W.
CALGARY ALBERTA
CANADA T3E-3N6

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us a poor photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de mauvaise qualité.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

THE UNIVERSITY OF ALBERTA

NUMERICAL AND ANALYTIC STUDIES OF LASER-PLASMA INTERACTIONS IN A SOLENOIDAL MAGNETIC FIELD

BY

© RICHARD DOUGLAS MILROY

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL ENGINEERING

EDMONTON, ALBERTA

FALL, 1978

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, "Numerical and Analytic Studies of Laser-Plasma Interactions in a Solenoidal Magnetic Field", submitted by Richard Douglas Milroy in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering.

Supervisor

Supervisor

External Examiner

Date: August 4, 1978

ABSTRACT

A two-dimensional computer simulation of a plasma in a solenoidal magnetic field is developed. This simulation is based on a magnetic flux shell model of the plasma. Effects of finite electrical conductivity and anisotropic heat conductivity are included. The computer code can optionally calculate the full radial dynamics of the problem or make the approximation of radial pressure balance.

The computer program has been used to do sample calculations on laser heated plasmas confined in a solenoid and on a θ-pinch reactor. The program has also been used for a numerical study of the hydrodynamic behaviour of laser heated gas target plasmas.

A theory for the beat frequency mixing of antiparallel laser beams in a homogeneous plasma is developed. This theory includes effects of ion mobility and electron-ion collisions. Based on this theory and MHD simulations of laser heated plasmas in solenoidal magnetic fields, a plasma parametric amplifier of infrared laser radiation is proposed. Analysis of this device indicates that it could be used to efficiently amplify infrared radiation and may also be capable of creating and amplifying extremely short pulses of infrared radiation.

ACKNOWLEDGEMENTS

The author wishes to express his thanks to Dr. C.E. Capjack and Dr. C.R. James who supervised this project. A special thanks is also extended to Dr. J.N. McMullin for the many helpful discussions and the substantial contributions he has made to this work.

A special thanks is also extended to Dr. A.A. Offenberger, Mr. L. Pitt, and Dr. A. Ng for their many stimulating discussions; and to Ms. Barbara J. Gallaiford for the speedy and accurate typing of this thesis.

The author is also thankful for the financial support provided by the National Research Council of Canada and the University of Alberta.

Most of all the author would like to thank his wife, Kay, for her patience and encouragement during the course of this work.

TABLE OF CONTENTS

CHAPTER PAGE

CHAPTER		PAGE

LIST OF FIGURES

FIGURE PAGE

FIGURE PAGE

FIGURE		PAGE

FIGURE		PAGE

FIGURE PAGE

FIGURE		PAGE
A.2	Flow chart illustrating two-step scheme	211
A.3	Position and indexing of numerical grid for calculation of laser power P, and absorbed energy E, relative to positioning and indexing of MHD variables	212

CHAPTER 1

Introduction

The physics of laser-plasma interactions is an important part of the controlled thermonuclear research program. While most of the effort in laser-plasma interaction research is directed towards the development of laser-pellet implosion fusion, a considerable effort is being devoted to the study of plasma dynamics and laser-plasma interactions in solenoids. In such a device a solenoidal magnetic field provides radial confinement of the plasma and inhibits thermal conduction in the radial direction. Dawson et al[1] have suggested that CO_2 laser heated plasma columns may offer an alternative approach to controlled thermonuclear fusion. In this scheme the plasma is heated through the absorption of a high intensity CO_2 laser beam that is directed down the solenoid axis. The beam is guided down the solenoid axis by an on-axis density minimum which is created by a high on-axis heating rate. Additional heating may be provided by an adiabatic compression achieved by increasing the strength of the solenoidal magnetic field after laser heating is terminated[2]. In the simplest concept the plasma confinement time is limited to the time required for the plasma to stream freely out of the solenoid ends. Solenoid lengths of about 1 km are required to achieve adequate confinement times[3]. It has been suggested that endloss could be reduced, and hence the device shortened by plugging the ends with solid material or high density gas. Heat loss to the ends through thermal conduction then becomes a serious problem.

Problems and ideas related to this concept of laser fusion are being investigated in several laboratories using short (< 1 m) solenoids. However, these experiments and reactor concepts are difficult to analyze theoretically. This is because a good theoretical understanding requires the simultaneous solution to the equations that govern many different but inter-related physical phenomena. Because of the complexity of this type of problem, computer simulations have become an essential tool for plasma physics research. Computer codes can incorporate many different but inter-related physical phenomena. This can be useful in testing simplified theories as well as providing a better understanding of experimental observations. In addition to explaining experimental results, computer models can predict unexpected phenomena, and be useful in the design of new experiments and reactor concepts.

The rapid advance of computer technology in the past few years has resulted in larger and faster computers being available to plasma researchers. This allows for the numerical solution to more sophisticated mathematical models leading to more realistic computer simulations.

In Chapter 2 of this thesis a two-dimensional computational model developed by Dr. J.N. McMullin and the author is described. This model has been designed to simulate the magnetohydrodynamic behaviour of a laser-heated plasma in a solenoidal magnetic field. An attempt has been made to make the computer code sufficiently versatile so that the one program can be used to study short experimental devices as well as reactor design concepts.

Electron thermal conductivity and fluid flow in a plasma in a strong magnetic field is markedly anisotropic. The thermal conductivity is much larger along the field lines than across it and plasma flow is dominantly along the field lines. Because of this, a large amount of numerical diffusion is experienced when normal differencing techniques are employed to solve the MHD equations. An extremely fine computational grid, resulting in an excessive amount of computer time needed to do long simulations, is then required in order to attain reasonable accuracy. This problem has been eliminated in the present model by transforming all of the equations into a moving non-orthogonal coordinate system defined by the magnetic field lines.

Another feature built into the present numerical model is the option of assuming radial pressure balance or calculating full radial dynamics. In many cases the time scales of interest are much longer than the time required for a magneto-acoustic wave to propagate across the solenoid radius. When this is true any pressure imbalance in the radial direction is quickly smoothed out and the assumption of radial pressure balance in a numerical model is valid. The main advantage of making this assumption is that it eliminates what can be a very restrictive stability condition. The present model treats diffusion terms implicitly and all other terms are treated explicitly. Consequently, the timestep size is limited by the Courant-Friedricks-Lewy condition[4]. If full radial dynamics are calculated the timestep size is limited by the following two conditions:

$$\Delta t \leq \text{Min} \left\{ \frac{\Delta R}{C_{ma}} \right\} \qquad (1.1)$$

$$\Delta t \leq \text{Min} \left\{ \frac{\Delta x}{C_s} \right\} \qquad (1.2)$$

where Δt is the timestep size, C_{ms} is the magneto-acoustic velocity, ΔR is the numerical grid size in the radial direction, C_s is the sound velocity, and Δx is the numerical grid size in the axial direction. For almost all simulations of plasmas in a solenoidal magnetic field condition (1.1) is much more restrictive than condition (1.2). The assumption of radial pressure balance eliminates condition (1.1) and timestep size is limited only by condition (1.2). For the simulations of long solenoids such as reactors this reduces the required number of calculations by several orders of magnitude.

Flux coordinates were used in a numerical model developed by Hertweck and Schneider[5] and used by Schneider[6] and Bodin et al.[7] in the study of end-loss from θ-pinches. However, their model assumed infinite electrical conductivity so that the effects of magnetic diffusion in the plasma were not included. Although this approximation was adequate for their studies, it breaks down completely at low temperatures during the early stages of heating of the plasma. Magnetic diffusion is also expected to affect the end-loss from long, high β reactors[8] even at very high temperatures so finite conductivity has been included in the present model. Other improvements in this model are (1) the inclusion of full radial dynamics using an implicit scheme for calculating the motion of flux surfaces; (2) the use of a direct (rather than iterative) method for advancing the equations when radial pressure balance is assumed; (3) the simplification of the equations of motion by assuming only minor distortion of the field lines.

Sample calculations, using this computational model, are presented in Chapter 3. The versatility of the code is demonstrated through the presentation of results from solenoids of lengh 5 cm, 1 m, and 1 km.

The code has also been used to provide numerical backup for experiments performed by Dr. A.A. Offenberger[9]. In Chapter 4 some results from a computer study of the hydrodynamics of gas target experiments are presented. In these simulations a CO_2 laser beam is assumed to be focussed into a semi-infinite slab of unmagnetized plasma. The laser heating of the plasma and resulting hydrodynamic expansion are modelled. Because the plasma is unmagnetized and the model uses a coordinate system based on the motion of magnetic flux lines some small alterations had to be made to the code for these simulations.

The results of Chapter 3 suggest that a laser heated plasma in a 1 m solenoid can have very long axial scale lengths. Simulations show that a plasma can be created with parameters ideal for the beat frequency mixing of antiparallel laser radiation. It is suggested in this thesis that this process could be used to create a very efficient broad-band amplifier of infared radiation[10] as well as a device capable of creating or amplifying extremely short bursts of infrared radiation.

Krupke[11] has demonstrated that a short pulse of laser radiation can be amplified through the absorption of energy from a longer pulse of laser radiation propagating antiparallel to the short pulse. The radiation in the longer, or pump, pulse has a slightly higher frequency than the short pulse. The process works as follows: a high intensity pulse of laser radiation is directed into a mixing medium, in this case CH_4 gas. A short pulse of lower frequency laser radiation is directed antiparallel to the original pulse. Energy is transferred from the high frequency pulse to the low frequency pulse through a Raman process. Thus the original pulse energy is depleted and the short pulse is amplified. If the process is fast enough most of the energy of the pump will be transferred to the front of the amplified pulse. In this way energy from a longer laser pulse is used to create an extremely high intensity short pulse. It is feasible that such a process could be used to efficiently create the short pulses of radiation needed for laser pellet fusion.

In Krupke's experiment neutral CH_4 gas was the mixing medium for 259 nm and 268 nm wavelength pulses of radiation from KrF lasers. In this thesis it is suggested that a plasma could be used as a mixing medium for infrared radiation in the 10.6 μm range. The mixing can take place between waves with a difference frequency close to the electron plasma frequency (a Raman process) or the ion-acoustic frequency (a Brillouin process).

Chapter 5 contains a review of some of the theory of the beat frequency mixing of antiparallel laser beams and extends this theory to provide a basis for the analysis of a device capable of amplifying ultra-short pulses of laser radiation. Chapter 6 contains a theoretical analysis of two proposed devices which rely on the beat frequency mixing of laser radiation in a plasma.

CHAPTER 2

A Two-Dimensional Magnetic Flux Shell Model for M.H.D. Simulations

This chapter describes a two-dimensional computational model developed by Dr. J.N. McMullin and the author to simulate the magneto-hydrodynamics of a cylindrically symmetric plasma with an imbedded solenoidal magnetic field. This type of numerical simulation is a valuable supplement to current laboratory experiments designed to test the feasibility of using a magnetically confined plasma column as a thermonuclear device. Numerical models are also the only way at present for evaluating proposed designs of fusion reactors and neutron sources based on linear plasma columns. Because the geometries of current experiments and reactor designs are basically the same, both can be simulated with a single code even though the dimensions and physical parameters vary over wide ranges.

A common feature of all experiments and reactor designs is the attainment of high electron temperatures (>40eV). Due to the resulting large electrical conductivity, the motion of plasma particles across magnetic field lines is strongly inhibited. Strong magnetic fields can therefore be used to restrain the plasma against expansion during heating as well as to inhibit loss of heat by conduction to the surrounding walls. The main losses of internal energy of these plasmas are convection and heat conduction along the field lines which join the confined plasma region to the outside world. If the magnetic field lines are used as coordinates, the motion of the plasma is mainly one-dimensional along these coordinates. Furthermore, there are no mixed derivatives in the parallel heat conduction term of the electron temperature equation so that heat conduction along field lines may be calculated free of errors due to numerical diffusion across coordinates. For these reasons, the numerical model uses magnetic field line, or flux coordinates, (s,x,t), where s(r,x,t) is the magnetic flux through a circle of radius r centered on the x-axis, instead of the regular cylindrical coordinates,(r,x,t).

2.1 The Hydrodynamic Model

The present model is based on the hydrodynamic equations given by Braginskii[12]. The plasma is approximated as a fully ionized two-temperature ideal magnetohydrodynamic fluid. The electrons and ions are assumed to have the same local number density and the same fluid velocity. Cylindrical symmetry is assumed in the derivation of all the equations. A solenoidal magnetic field is assumed to permeate the plasma although it may have a vanishing field strength. The magnetic field is assumed to diffuse due to a classical electrical resistivity with joule heating being absorbed by the electrons only. The strong anisotropy of thermal conduction perpendicular and parallel to the magnetic field lines is accounted for. Electrons and ions are assumed to exchange energy at the classical equipartition rate. An artifical viscosity is introduced to numerically handle shock waves which may be generated by a strong expansion. The artificial viscosity has the effect of spreading a shock front over several mesh points as described by Richtmeyer and Morton[4]. The model allows for the heating of electrons through the inverse Bremsstrahlung absorption of a laser beam propagating down the solenoid axis.

The equations used are listed below in cylindrical coordinates (r,x,t). In these equations n_o is the electron number density, m_i is the ion mass, $\rho = m_i n_o$ is the mass density, v_x and v_r are the axial and radial components of fluid velocity, T_e and T_i are the electron and ion temperatures, and $P = n_o k_B (T_e + T_i)$ is the scalar pressure. B_x and B_r are the axial and radial components of the magnetic field. The total derivative, $\frac{d}{dt}$, is the convective derivative:

$$\frac{d}{dt} = \frac{\partial}{\partial t} + v_x \frac{\partial}{\partial x} + v_r \frac{\partial}{\partial r}$$

2.1.1 Continuity Equation:

$$\frac{dn_o}{dt} + n_o \vec{\nabla} \cdot \vec{v} = 0$$

or in cylindrical coordinates:

$$\frac{dn_o}{dt} + n_o \left[\frac{\partial v_x}{\partial x} + \frac{1}{r} \frac{\partial}{\partial r} (r\, v_r) \right] = 0 \qquad (2.1)$$

2.1.2 Momentum Equation:

$$\rho \frac{d\vec{v}}{dt} = - \nabla P + \frac{1}{c} (\vec{J} \times \vec{B}) - \frac{\partial q_r}{\partial r} \hat{r} - \frac{\partial q_x}{\partial x} \hat{x}$$

where q_r and q_x represent artificial viscosities in the radial and axial directions respectively.

$$q_r = \begin{cases} a_r^2 \rho \left(\frac{\partial v_r}{\partial r} \right)^2 & \frac{\partial v_r}{\partial r} < 0 \\ 0 & \frac{\partial v_r}{\partial r} \geq 0 \end{cases}$$

a_r is a variable having the dimensions of length and is chosen to spread a shock over several radial mesh points.

$$q_x = \begin{cases} a_x^2 \rho \left(\frac{\partial v_x}{\partial x} \right)^2 & \frac{\partial v_x}{\partial x} < 0 \\ 0 & \frac{\partial v_x}{\partial x} \geq 0 \end{cases}$$

a_x is a variable having the dimensions of length and is chosen to spread a shock over several axial mesh points. Splitting the acceleration equation into axial and radial components and eliminating $\vec{J}$ with $\vec{\nabla}\times\vec{B} = (4\pi/c)\ \vec{J}$ yields:

$$\rho \frac{dv_x}{dt} = - \frac{\partial P}{\partial x} - \frac{B_r}{4\pi} \left(\frac{\partial B_r}{\partial x} - \frac{\partial B_x}{\partial r} \right) - \frac{\partial q_x}{\partial x} \tag{2.2}$$

$$\rho \frac{dv_r}{dt} = - \frac{\partial P}{\partial r} + \frac{B_x}{4\pi} \left(\frac{\partial B_r}{\partial x} - \frac{\partial B_x}{\partial r} \right) - \frac{\partial q_r}{\partial r} \tag{2.3}$$

2.1.3 Temperature Equation: (ε = e or i for electrons or ions respectively)

$$n_o \frac{dT_\varepsilon}{dt} - (\gamma-1)\ T_\varepsilon \frac{dn_o}{dt} = n_o \frac{dT_\varepsilon}{dt}\Big|_{coll} + \frac{(\gamma-1)}{K_B}[n_o \epsilon_L - \nabla\cdot Q_\varepsilon + H_{jul}] \tag{2.4}$$

The first term on the right gives the rate of energy transfer between electrons and ions by collisions. In the electron temperature equation this term may be approximated as:

$$\frac{dT_e}{dt}\Big|_{coll} = \frac{1}{\tau_{eq}} (T_i - T_e) \tag{2.5}$$

where τ_{eq} is the appropriate collision time. The same term with opposite sign appears in the ion equation. The second term represents the inverse Bremsstrahlung absorption of laser energy while the third term represents thermal conduction. The last term, which accounts for electron heating due to magnetic field diffusion can be expressed as:

$$H_{jul} = \frac{J_{\perp}^2}{\sigma_{\perp}} .$$

where $\sigma_{\perp}$ is the perpendicular component of the electrical conductivity tensor. Only the perpendicular component of the electrical conductivity tensor is needed since the current is purely azimuthal when the magnetic field is solenoidal. The heat flux vector $\vec{Q}_{\varepsilon}$ is given by:

$$\vec{Q}_{\varepsilon} = - K_{\varepsilon} \cdot \nabla T_{\varepsilon}$$

where K_{ε} is the heat conductivity tensor. Shock heating, which can be accounted for through the artificial viscosity term, has been neglected in the present treatment.

2.1.4 Magnetic Field Equations:

Equations for the magnetic field components can be written as:

$$B_r = - \frac{1}{r} \int_0^r \frac{\partial B_x}{\partial x} r' \, d r' \qquad (2.6)$$

$$\frac{\partial B_x}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} [r(v_x B_r - v_r B_x)] + \frac{c^2}{4\pi r} \frac{\partial}{\partial r} [\frac{r}{\sigma_{\perp}} (\frac{\partial B_x}{\partial r} - \frac{\partial B_r}{\partial x})] \qquad (2.7)$$

Equation (2.6) can be derived from $\vec{\nabla} \cdot \vec{B} = 0$ while equation (2.7) is obtained from Maxwells equations and Ohm's Law,

$$\vec{J} = \sigma(\vec{E} + \frac{1}{c} \vec{v} \times \vec{B})$$

2.1.5 Laser Absorption and Propagation:

The electron heating rate due to the inverse Bremsstrahlung absorption of laser energy used in equation (2.4) can be written as:

$$n_o \epsilon_L = K_a I_l(r,x,t) \qquad (2.8)$$

where I_l is the laser beam intensity and K_a is the absorption coefficient. The laser beam has a given radial profile, Gaussian for most cases, which is assumed to be unaltered as the beam propagates down the solenoid axis. The total beam power P_L where:

$$P_L(x,t) = 2\pi \int_0^\infty I_l(r,x,t)\, r\, d\, r \qquad (2.9)$$

obeys the transfer equation:

$$\frac{1}{c}\frac{\partial}{\partial t} P_L + \frac{\partial}{\partial x} P_L = - \bar{K}_a P_L \qquad (2.10)$$

c is the speed of light and $\bar{K}_a$ is the absorption coefficient averaged over radial profile with the weights at each radial position being proportional to the beam intensity at that point. This scheme reflects the fact that a light ray is refracted through regions of varying absorption as it propagates through the plasma.

2.1.6 Transport Coefficients:

Values for the transport coefficients that have been used in this model have been taken from the Revised NRL Plasma Formulary[13]. The formula connecting the low and high magnetic field limits for perpendicular thermal conductivity comes from Hain et al[14]. In the following temperature is measured in eV and k_B = 1.6x10^{-12} ergs/eV.

Thermal Conduction Parallel to $\vec{B}$

$$k^e_{11} = 3.1\times10^9 \frac{T_e^{5/2}}{\ell n\Lambda} \left(\frac{\text{ergs}}{\text{eV sec cm}} \right)$$

$$k_{11}^{i} = 1.25\times10^{8}\ \frac{T_i^{5/2}}{\ell n\Lambda}\ \left(\frac{\text{ergs}}{\text{eV sec cm}}\right)$$

Thermal Conduction Perpendicular to $\vec{B}$

$$k_{\perp}^{\varepsilon} = \frac{\delta_1^{\varepsilon}\ T_{\varepsilon}^{5/2}}{\left(1 + \delta_2^{\varepsilon}\ \frac{T_{\varepsilon}^{3}B^{2}}{n_o^{2}}\right)}\qquad \left(\frac{\text{ergs}}{\text{eV sec cm}}\right)$$

where

$$\delta_1^{e} = \frac{3.1\times10^{9}}{\ell n\Lambda}\qquad \delta_1^{i} = \frac{1.25\times10^{8}}{\ell n\Lambda}$$

$$\delta_2^{e} = \frac{2.45\times10^{25}}{(\ell n\Lambda)^2}\qquad \delta_2^{i} = \frac{7.8\times10^{22}}{(\ell n\Lambda)^2}$$

In the limit of large magnetic fields, $\omega_{c\varepsilon} >> \nu_{\varepsilon}$, (where $\omega_{c\varepsilon}$ is the cyclotron frequency of species ε, and ν_{ε} is the collision frequency of species ε)

$$k_{\perp}^{\varepsilon} \approx \frac{\delta_1^{\varepsilon}}{\delta_2^{\varepsilon}}\ \frac{n_o^2}{B^2\sqrt{T_{\varepsilon}}}$$

In the limit of small magnetic fields, $\omega_{c\varepsilon} << \nu_{\varepsilon}$,

$$k_{\perp}^{\varepsilon} = \delta_1^{\varepsilon}\ T_{\varepsilon}^{5/2}$$

Electrical Conductivity

$$\sigma_{\perp} = 8.7\times10^{13}\ \frac{T_e^{3/2}}{\ell n\Lambda}\qquad \text{sec}^{-1}$$

Electron-Ion Equipartition of Energy

$$\tau_{eq} = \frac{1}{\nu_{ei}}$$

$$\nu_{ei} = 3.25\times10^{-9} \frac{n_o \ell n\Lambda}{T_e^{3/2}} \quad sec^{-1}$$

Laser Absorption Coefficient

The value given by Johnston and Dawson[15] is used and for λ=10.6μm

$$K_a = \frac{9.74\times10^{-36} n_o^2}{T_e^{3/2}} \ell n\Lambda$$

where

$$\Lambda = \min\{2.3 \cdot T_e^{3/2}, 12 \cdot T_e\}$$

2.2 The Magnetic Flux Coordinate System

A number of different numerical schemes may be used to self-consistently solve the partial differential equations in Section 2.1. Since energy transport due to either particle convection or thermal conduction is strongly anisotropic, with the dominant energy transport being along the magnetic field lines, any such scheme will introduce considerable numerical diffusion in cases where the magnetic field lines are not parallel to the x axis. In the present model the differential equations are transformed into a time dependent non-orthogonal coordinate system defined by the magnetic field lines. In this coordinate system energy transport occurs dominantly along one of the coordinates and the problem of large numerical diffusion is eliminated.

2.2.1 Definition of the Coordinate System

The transformation of variables from cylindrical coordinates (r,x,t) to flux coordinates (s,x',t') is defined below.

$$
\begin{aligned}
s &= S(r,x,t) = 2\pi \int_0^r B_x(\tilde{r},x,t)\,\tilde{r}\,d\tilde{r} \\
x' &= x \qquad\qquad (2.11) \\
t' &= t
\end{aligned}
$$

The corresponding inverse transform can be written as:

$$
\begin{aligned}
r &= R(s,x',t) \\
x &= x' \qquad\qquad (2.12) \\
t &= t'
\end{aligned}
$$

The radius, $R(s,x,t)$, of a flux surface of constant s now becomes a plasma variable from which the magnetic field components can be determined.

2.2.2 Transformation of Derivations

Relations governing the transformation of derivatives between the two coordinate systems will now be obtained.

$$\begin{bmatrix} ds \\ dx' \\ dt' \end{bmatrix} = X \begin{bmatrix} dr \\ dx \\ dt \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} dr \\ dx \\ dt \end{bmatrix} = Y \begin{bmatrix} ds \\ dx' \\ dt' \end{bmatrix} \tag{2.13}$$

where

$$X = \begin{bmatrix} \partial_r S & \partial_x S & \partial_t S \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} \partial_s R & \partial_{x'} R & \partial_{t'} R \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.14}$$

From equation (2.13) $X = Y^{-1}$ and

$$Y^{-1} = \begin{bmatrix} \frac{1}{\partial_s R} & -\frac{\partial_{x'} R}{\partial_s R} & -\frac{\partial_{t'} R}{\partial_s R} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.15}$$

A term wise comparison of X and Y^{-1} yields

$$\partial_r S = \frac{1}{\partial_s R}, \partial_x S = -\frac{\partial_{x'} R}{\partial_s R}, \text{ and } \partial_t S = -\frac{\partial_{t'} R}{\partial_s R}$$

By using these relations, the partial derivations transform as follows:

$$\frac{\partial}{\partial t} \rightarrow \frac{\partial}{\partial t'} + \frac{\partial S}{\partial t}\,\frac{\partial}{\partial s} = \frac{\partial}{\partial t'} - \frac{\partial_{t'} R}{\partial_s R}\,\frac{\partial}{\partial s}$$

$$\frac{\partial}{\partial x} \rightarrow \frac{\partial}{\partial x'} + \frac{\partial S}{\partial x}\,\frac{\partial}{\partial s} = \frac{\partial}{\partial x'} - \frac{\partial_{x'} R}{\partial_s R}\,\frac{\partial}{\partial s} \qquad (2.16)$$

$$\frac{\partial}{\partial r} \rightarrow \frac{\partial S}{\partial r}\,\frac{\partial}{\partial s} = \frac{1}{\partial_s R}\,\frac{\partial}{\partial s}$$

The right hand side of equations (2.16) is now a function of the new variables. Some other differential operators and quantities that will be required to express the MHD equations in the new coordinate system are derived below.

$$\frac{d}{dt} = \frac{\partial}{\partial t} + \vec{v}\cdot\vec{\nabla} \rightarrow \frac{\partial}{\partial t'} + \frac{\partial S}{\partial t}\frac{\partial}{\partial s} + v_x \frac{\partial}{\partial x'} + v_x \frac{\partial S}{\partial x}\frac{\partial}{\partial s} + v_r \frac{\partial S}{\partial r}\frac{\partial}{\partial s}$$

$$= \frac{\partial}{\partial t'} + v_x \frac{\partial}{\partial x'} + \left(\frac{\partial S}{\partial t} + \vec{v} \cdot \vec{\nabla} S\right) \frac{\partial}{\partial s}$$

or

$$\frac{d}{dt} \rightarrow \frac{\partial}{\partial t'} + v_x \frac{\partial}{\partial x'} + \frac{dS}{dt}\frac{\partial}{\partial s} \qquad (2.17)$$

For any vector $\vec{V}$ defined in cylindrical coordinates

$$\vec{\nabla} \cdot \vec{V} = \frac{1}{r}\frac{\partial}{\partial r}(r\, V_r) + \frac{\partial V_x}{\partial x} \qquad (2.18)$$

By using equations (2.16), this transforms to:

$$\vec{\nabla} \cdot \vec{V} = \partial_{x'} V_x - \frac{\partial_{x'} R}{\partial_s R} \partial_s V_x + \frac{1}{R \partial_s R} \partial_s (R V_r) \tag{2.19}$$

An expression for v_r can be obtained.

$$v_r = \frac{dR}{dt} = \frac{\partial R}{\partial t'} + v_x \frac{\partial R}{\partial x'} + \frac{dS}{dt} \frac{\partial R}{\partial s} \tag{2.20}$$

Equations (2.17) and (2.20) both contain the factor $\frac{dS}{dt}$. This factor can be expressed in terms of the magnetic field components. In cylindrical coordinates:

$$\frac{dS}{dt} = \frac{\partial S}{\partial t} + v_r \frac{\partial S}{\partial r} + v_x \frac{\partial S}{\partial x} \tag{2.21}$$

From equation (2.11) and $\vec{\nabla} \cdot \vec{B} = 0$, it can be shown that:

$$\frac{\partial S}{\partial x} = - 2\pi r B_r \tag{2.22}$$

From equation (2.11):

$$\frac{\partial S}{\partial r} = 2\pi r B_x \tag{2.23}$$

From equations (2.11) and (2.7):

$$\frac{\partial S}{\partial t} = 2\pi r (v_x B_r - v_r B_x) + \frac{c^2}{2\sigma_\perp} r \left(\frac{\partial B_x}{\partial r} - \frac{\partial B_r}{\partial x} \right) \tag{2.24}$$

By substituting equations (2.21), (2.22), and (2.24) into equation (2.20):

$$\frac{dS}{dt} = \frac{c^2}{2\sigma_\perp} r \left(\frac{\partial B_x}{\partial r} - \frac{\partial B_r}{\partial x} \right) \tag{2.25}$$

2.2.3 Dimensionless Variables

Numerical techniques will be employed to find approximate solutions to the magnetohydrodynamic equations in the magnetic flux coordinate system. It has been found that the algebra can be greatly simplified and computer time can be significantly reduced if certain small terms are neglected in the magnetohydrodynamic equations. Such small terms can most easily be identified if the M.H.D. equations are expressed in terms of a set of dimensionless variables. By using dimensionless variables in the computer program the number of constant multipliers in the difference equations is reduced which in turn reduces the computer time required for a solution. In converting equations to dimensionless units, dimensional quantities Q are replaced by $Q_o\tilde{Q}$ where Q_o is a convenient unit and $\tilde{Q}$ is the dimensionless variable. The four basic quantities used to define the dimensionless system are defined below.

(i)	L_o	-	axial length unit	$x' \to L_o \tilde{x}$
(ii)	R_o	-	radial length unit	$r' \to R_o \tilde{r}$
(iii)	N_o	-	number density	$n_o \to N_o \tilde{N}$
(iv)	T_o	-	temperature	$T_{e,i} \to T_o \tilde{T}_{e,i}$

The other basic units that will be used are defined in terms of the above four variables and are listed below.

(i)	P_o	$= N_o k_B T_o$	-	Pressure
(ii)	B_o	$= (8\pi P_o)^{1/2}$	-	Magnetic field
(iii)	v_o	$= (k_B T_o/m_i)^{1/2}$	-	Axial velocity

(iv) $t_o = L_o/v_o$ - Time $t' = t_o\tilde{t}$

(v) $A_o = \pi R_o^2$ - Area

(vi) $S_o = A_o B_o$ - Magnetic flux

(vii) $E_o = N_o k_B T_o L_o A_o$ - Energy

(viii) $P_{wo} = N_o k_B T_o A_o v_o$ - Power

2.2.4 Expansion Parameter ϵ_1

The length of a typical solenoidal device that we are interested in modelling will invariably be much greater than its radius. As a result $L_o >> R_o$ and a small parameter $\epsilon_1 \equiv R_o/L_o$ can be defined. When the MHD equations are expressed in terms of dimensionless units in the magnetic flux coordinate system, many terms appear that are of order ϵ_1^2. In the present treatment such terms have been neglected. Neglecting terms of order ϵ_1^2 is equivalent to assuming that the slope of a field line with respect to the axis of a solenoid is a small quantity ϵ and that terms of the order of ϵ_1^2 can be neglected. For a magnetic field in a typical solenoid, $\epsilon_1 << 1$ except for small regions near each end where the field lines are diverging. The axial length of these regions is of the order of the solenoid diameter which for most experiments and all reactor designs is a small fraction of the total plasma. In laser heating experiments in which a bleaching wave propagates through the plasma, the approximation should remain valid if any one of the four following conditions holds:

(1) $B^2/8\pi >> P$

(2) Absorption length, $L_{abs} >> R_{las}$, Radius of Laser Beam

(3) Bleaching Wave Velocity, $V_{bw} >> V_{ma}$, Magneto-Acoustic Velocity

(4) $c^2/(4\pi V_{ma} R_{las}) >> \sigma$, Electrical Conductivity.

Condition (1) ensures that the magnetic pressure will prevent significant expansion of the plasma and hence distortion of the field lines. If condition (1) does not hold, then it is reasonable to assume that the region being heated and still expanding is on the order of $L = \min(V_{bw}\Delta t, L_{abs})$ where $\Delta t = R_{las}/V_{ma}$ is the approximate expansion time of the plasma under uneven radial heating. Since the displacement of a field line is on the order of the radius of the heated region, R_{las}, the slope of the field lines is approximately $\epsilon_1 = R_{las}/L$. Either of conditions (2) or (3) ensure that $\epsilon_1 << 1$. When condition (4) holds, smoothing of the magnetic field by diffusion dominates over distortion of the field by fluid convection.

2.2.5 Transformation of Derivatives and Associated Differential Operators to Dimensionless Variables

The transformation of the variable S, as defined in equation (2.11) into dimensionless form yields:

$$\tilde{S} = 2 \int_0^{\tilde{r}} \tilde{B}_{\tilde{x}} \, \tilde{r}' \, d\tilde{r}' \qquad (2.26)$$

It then follows that:

$$\frac{\partial \tilde{S}}{\partial \tilde{r}} = 2\tilde{B}_{\tilde{x}}\tilde{R} = \frac{1}{\partial_{\tilde{s}}\tilde{R}}$$

or

$$\tilde{B}_{\tilde{x}} = \frac{1}{2\tilde{R}\,\partial_{\tilde{s}}\tilde{R}} \qquad (2.27)$$

The subscript $\tilde{x}$ will be dropped on $B_{\tilde{x}}$ in subsequent analysis. The variable D is defined as follows:

$$D \equiv \frac{dS}{dt} \qquad (2.28)$$

The variable D is converted into dimensionless variables by using equation (2.25) and dropping terms of order ϵ_1^2.

$$D = \frac{c^2}{2\sigma_\perp} B_o \tilde{R} \frac{\partial \tilde{B}}{\partial \tilde{r}} = D_o \tilde{D} \qquad (2.29)$$

where

$$D_o \equiv \frac{c^2 B_o}{2\sigma_\perp}$$

Two other quantities that will be frequently used in subsequent analysis are:

$$D \frac{\partial}{\partial s} = \frac{V_o}{L_o} \epsilon_2 \tilde{D} \frac{\partial}{\partial \tilde{s}}$$

$$\frac{\partial D}{\partial s} = \frac{V_o}{L_o} \frac{\partial}{\partial \tilde{s}} [\epsilon_2 \tilde{D}] \qquad (2.30)$$

where

$$\epsilon_2 = \frac{c^2 L_o}{2\pi\sigma_\perp R_o^2 v_o}$$

The magnetohydrodynamic equations defined in Section 2.1 can be converted directly from cylindrical coordinates to magnetic flux coordinates in dimensionless variables by using the differential operators in equations (2.16) to (2.17) expressed in terms of dimensionless variables. The operators may be shown to take the following form.

$$\frac{\partial}{\partial t} \rightarrow \frac{v_o}{L_o}\left[\partial_{\tilde{t}} - \frac{\partial_{\tilde{t}}\tilde{R}}{\partial_{\tilde{s}}\tilde{R}}\,\partial_{\tilde{s}}\right]$$

$$\frac{\partial}{\partial x} \rightarrow \frac{1}{L_o}\left[\partial_{\tilde{x}} - \frac{\partial_{\tilde{x}}\tilde{R}}{\partial_{\tilde{s}}\tilde{R}}\,\partial_{\tilde{s}}\right] \tag{2.31}$$

$$\frac{\partial}{\partial r} \rightarrow \frac{1}{R_o}\left[\frac{1}{\partial_{\tilde{s}}\tilde{R}}\,\partial_{\tilde{s}}\right]$$

$$\frac{d}{dt} \rightarrow \frac{v_o}{L_o}\left[\partial_{\tilde{t}} + \tilde{v}\,\partial_{\tilde{x}} + \varepsilon_2 D\,\partial_{\tilde{s}}\right]$$

where here and in all subsequent analysis the $\tilde{x}$ subscript will be dropped from $\tilde{v}_{\tilde{x}}$. Equation (2.19) can be expressed in dimensionless variables as:

$$\vec{\nabla}\cdot\vec{V} = \frac{V_o}{L_o}\left[\partial_{\tilde{x}}\tilde{V}_x - \frac{\partial_{\tilde{x}}\tilde{R}}{\partial_{\tilde{s}}\tilde{R}}\,\partial_{\tilde{s}}\tilde{V}_x\right] + \frac{V_o}{R_o}\left[\frac{1}{\tilde{R}\,\partial_{\tilde{s}}\tilde{R}}\,\partial_{\tilde{s}}(\tilde{R}\,\tilde{V}_r)\right] \tag{2.32}$$

An expression v_r, defined in equation (2.19), can be written in terms of dimensionless variables as:

$$v_r = v_o\,\frac{R_o}{L_o}\left[\partial_{\tilde{t}}\tilde{R} + \tilde{v}\,\partial_{\tilde{x}}\tilde{r} + \varepsilon_2\tilde{D}\,\partial_{\tilde{s}}\tilde{R}\right] \tag{2.33}$$

2.3 Transformation of M.H.D. Equations to the Dimensionless Magnetic Flux Coordinate System

The equations which define the magnetohydrodynamic model of Section 2.1 will now be converted to the coordinate system defined in Section 2.2.

2.3.1 Continuity Equation

By using the transformations indicated in equations (2.31) in equation (2.1) the following expression for the continuity equation in dimensionless magnetic flux coordinates is obtained:

$$\partial_t(N\,R\,\partial_s R) + \partial_x(v\,N\,R\,\partial_s R) + \partial_s(\varepsilon_2 D\,N\,R\,\partial_s R) = 0 \quad (2.34)$$

All variables in equation (2.34) are dimensionless. For the sake of clarity the ∿ over the variables has been dropped in this equation.

2.3.2 Momentum Equation

In converting equations (2.2) and (2.3) to dimensionless form the variable B_r may be written in terms of B_x and $R(s,x,t)$. If s and t are held constant, lines defined by $R(s,x,t)$ and x define magnetic field lines. That is

$$\frac{\partial R}{\partial x} = \frac{B_r}{B_x}$$

Thus,

$$B_r = B_x \frac{\partial R}{\partial x}$$

or in terms of dimensionless variables

$$B_r = \epsilon_1 B_o \tilde{B} \frac{\partial \tilde{R}}{\partial \tilde{x}} \tag{2.35}$$

Other relationships that will be used are

$$\rho = m_1 N_o \tilde{N} = N_o \left(\frac{k_B T_o}{v_o^2}\right) \tilde{N}$$

$$v_r = \left(\frac{v_o R_o}{L_o}\right) \tilde{v}_r$$

$$\tilde{v}_r = \partial_{\tilde{t}} \tilde{R} + \tilde{v} \partial_{\tilde{x}} \tilde{R} + \epsilon_2 \tilde{D} \partial_{\tilde{s}} \tilde{R} \tag{2.36}$$

$$\tilde{P} = \tilde{N} \tilde{T}$$

By using the above relationships plus the transformations given by equations (2.31), equation (2.3) transforms to

$$N_o k_B T_o \frac{R_o}{L_o^2} \tilde{N}(\partial_{\tilde{t}} + \tilde{v}\partial_{\tilde{x}} + \epsilon_2 \tilde{D}\partial_{\tilde{s}}) \tilde{v}_r = - \frac{N_o k_B T_o}{R_o} \frac{1}{\partial_{\tilde{s}} \tilde{R}} \partial_{\tilde{s}} (\tilde{P} + \tilde{q}_r)$$

$$- \frac{B_o^2 \tilde{B}}{4\pi R_o} \frac{1}{\partial_{\tilde{s}} \tilde{R}} \partial_s \tilde{B} \tag{2.37}$$

where the term $\frac{\partial B_r}{\partial x}$ has been neglected since it is of order ϵ_1^2 smaller than $\frac{\partial B_x}{\partial r}$, and $\tilde{q}_r$ is the appropriate expression in dimensionless variables for the artificial viscosity.

$$\tilde{q}_r = \begin{cases} (\tilde{a}_r \Delta R)^2 \, \epsilon_1^2 \, \tilde{N} \left(\frac{1}{\partial_{\tilde{s}} \tilde{R}} \partial_{\tilde{s}} \tilde{v}_r\right)^2 & \left(\frac{1}{\partial_{\tilde{s}} \tilde{R}} \partial_{\tilde{s}} \tilde{v}_r\right) < 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\tilde{a}_r$ is a dimensionless constant between 1.5 and 2.0 and ΔR is the radial grid spacing in the numerical calculations. Equation (2.37) can be re-written as

$$(\partial_t + v\partial_x + \epsilon_2 D \, \partial_s) \, v_r = - \frac{1}{N\epsilon_1^{\,2}} \frac{1}{\partial_s R} \partial_s (P + B^2 + q_r) \qquad (2.38)$$

where all variables are dimensionless and the $\sim$'s have been dropped for clarity in this equation.

The equation for axial momentum, equation (2.2), is converted to dimensionless variables in a similar manner. By using the transformations given by equations (2.31):

$$\frac{N_o k_B T_o}{L_o} \tilde{N}(\partial_{\tilde{t}} + \tilde{v}\partial_{\tilde{x}} + \epsilon_2 \tilde{D}\partial_{\tilde{s}})\tilde{v} = - \frac{N_o k_B T_o}{L_o} \frac{\partial}{\partial \tilde{x}} (\tilde{P} + \tilde{q}_x)$$

$$+ \frac{N_o k_B T_o}{L_o} \frac{\partial_{\tilde{x}} \tilde{R}}{\partial_{\tilde{s}} \tilde{R}} \partial_{\tilde{s}} \tilde{P} + \frac{R_o}{L_o} \frac{\partial_{\tilde{x}} \tilde{R} B_o^2 \tilde{B}}{4\pi R_o} \frac{1}{\partial_{\tilde{s}} \tilde{R}} \partial_{\tilde{s}} \tilde{B}$$

(2.39)

where the term $\frac{\partial B_r}{\partial x}$ has been neglected since it is of order $\epsilon_1^{\,2}$ smaller than $\frac{\partial B_x}{\partial r}$, and $\tilde{q}_x$ is the appropriate expression for the artificial viscosity.

$$\tilde{q}_x = \begin{cases} (\tilde{a}_x \Delta x)^2 \tilde{N} \left(\frac{\partial \tilde{v}}{\partial x} \right)^2 & \frac{\partial \tilde{v}}{\partial x} < 0 \\ 0 & \frac{\partial \tilde{v}}{\partial x} \geq 0 \end{cases}$$

where $\tilde{a}_x$ is a dimensionless constant between 1.5 and 2.0 and Δx is the axial grid spacing used in the numerical calculations. Equation (2.39) can be simplified to:

$$N(\partial_t + v\partial_x + \varepsilon_2 D\partial_s)\, v = - \frac{\partial}{\partial x} (P+q_x) + \frac{\partial_x R}{\partial_s R}\, \partial_s (P+B^2) \qquad (2.40)$$

where all variables are dimensionless and the ~'s have been dropped for clarity in this equation.

2.3.3 Tempefature Equation

The temperature equation for either species (ε = e or i for electrons or ions) in cylindrical coordinates is given by equation (2.4):

$$n_o \frac{dT_\varepsilon}{dt} - (\gamma-1)\, T_\varepsilon \frac{dn_o}{dt} = \frac{(\gamma-1)}{k_B} \vec{\nabla} \cdot \vec{Q}_\varepsilon + \frac{n_o(T_{\varepsilon'} - T_\varepsilon)}{\tau_{eq}}$$

$$+ \delta_{e\varepsilon} \frac{(\gamma-1)}{k_B} n_o \varepsilon_L + \delta_{e\varepsilon} \frac{\gamma-1}{k_B} H_{jul} \qquad (2.4)$$

where ε' = e if ε = i, ε' = i if ε = e, $\delta_{e\varepsilon} = 0$ if ε = i, and $\delta_{e\varepsilon} = 1$ if ε=e.

The most difficult term to resolve in equation (2.4) is the heat flow term. In the evaluation of an expression for this term ε will be dropped in order to make notation less cumbersome. A technique by which the heat flùx in the radial and x-directions may be evaluated is to consider an orthogonal coordinate system which is aligned with the magnetic field as depicted in Figure 2.1. In this figure the unit vectors $\hat{\varepsilon}_1$ and $\hat{\varepsilon}_2$ are defined as:

$$\hat{e}_1 = \hat{x} \cos\theta + \hat{r} \sin\theta$$

$$\hat{e}_2 = - \hat{x} \sin\theta + \hat{r} \cos\theta$$

Then energy flow due to thermal conduction can then be written as:

$$\vec{Q} = - k_{11}[\vec{\nabla}T \cdot \hat{e}_1] \hat{e}_1 - k_{\perp}[\vec{\nabla}T \cdot \hat{e}_2] \hat{e}_2$$

where k_{11} and $k_{\perp}$ are the thermal conductivities parallel and perpendicular to the magnetic field respectively, and are expressed in units of [ergs/(sec eV cm)]. Other terms in the above expression can be written as:

$$\vec{\nabla}T = \frac{\partial T}{\partial r} \hat{r} + \frac{\partial T}{\partial x} \hat{x}$$

$$\vec{\nabla}T \cdot \hat{e}_1 = \frac{\partial T}{\partial r} \sin\theta + \frac{\partial T}{\partial x} \cos\theta$$

$$\vec{\nabla}T \cdot \hat{e}_2 = \frac{\partial T}{\partial r} \cos\theta - \frac{\partial T}{\partial x} \sin\theta$$

The heat flow $\vec{Q}$ can then be d vided into a component in the radial direction, $Q_r \hat{r}$, and a component in the axial direction, $Q_x \hat{x}$.

$$\vec{Q} = Q_r \hat{r} + Q_x \hat{x} \qquad (2.41)$$

where

$$Q_r = - k_{11}[\partial_r T\sin\theta+\partial_x T\cos\theta]\sin\theta - k_\perp[\partial_r T\cos\theta-\partial_x T\sin\theta]\cos\theta$$

(2.42)

$$Q_x = - k_{11}[\partial_r T\sin\theta+\partial_x T\cos\theta]\cos\theta+ k_\perp[\partial_r T\cos\theta-\partial_x T\sin\theta]\sin\theta$$

The sine and cosine terms may be evaluated by considering Figure 2.2. If terms of order $\epsilon_1^{\ 2}$ are neglected,

$$\cos\theta \approx 1$$

(2.43)

$$\sin\theta \approx \epsilon_1 \, \partial_{\tilde{x}} \tilde{R}$$

Equations (2.41) to (2.43) can now be inserted into (2.32) to give an expression for $(\vec{\nabla}\cdot\vec{Q})$ in terms of dimensionless magnetic flux variables. In this step it is convenient to divide $(\vec{\nabla}\cdot\vec{Q})$ into two parts, $(\vec{\nabla}\cdot\vec{Q})_{11}$ which is a term accounting for k_{11}, and $(\vec{\nabla}\cdot\vec{Q})_\perp$ which is a term involving $k_\perp$.

$$\vec{\nabla}\cdot\vec{Q} = (\vec{\nabla}\cdot\vec{Q})_{11} + (\vec{\nabla}\cdot\vec{Q})_\perp \qquad (2.44)$$

After some algebraic manipulation and dropping terms of order ϵ_1^2

$$(\nabla \cdot Q)_{11} = - \frac{T_o}{L_o^2} \{ \frac{1}{\tilde{R}\partial_s \tilde{R}} \partial_{\tilde{x}} [(k_{11} \partial_{\tilde{x}} \tilde{T})(\tilde{R}\partial_{\tilde{s}} \tilde{R})]\} ,$$

(2.45)

$$(\nabla \cdot Q)_{\perp} = - \frac{T_o}{R_o^2} \frac{1}{\tilde{R}\partial_{\tilde{s}} \tilde{R}} \partial_{\tilde{s}} (\frac{\tilde{R}}{\partial_{\tilde{s}} \tilde{R}} k_{\perp} \partial_{\tilde{s}} \tilde{T})$$

The diffusion of a magnetic field through a plasma will lead to an ohmic heating of the electrons in the plasma. This is accounted for by the term H_{jul} in equation (2.4). An expression for this term is evaluated below.

$$H_{jul} = \vec{J} \cdot (\vec{E} + \frac{\vec{v} \times \vec{B}}{c})$$

where $\vec{J}$ is the induced electrical current, $\vec{E}$ is the electric field, and $\vec{v}$ is the fluid velocity. From Ohm's Law and Maxwell's equations the above expression can be written as:

$$\dot{H}_{jul} = \frac{J^2}{\sigma_{\perp}} = (\frac{c}{4\pi})^2 \frac{1}{\sigma_{\perp}} (\vec{\nabla} \times \vec{B})^2$$

where $\sigma_{\perp}$ is the electrical conductivity perpendicular to the magnetic field. For solenoidal fields $\vec{J}$ has a component in the $\hat{\phi}$ direction only. If terms of order ϵ_1^2 are neglected

$$\vec{\nabla} \times \vec{B} = - \frac{B_o}{R_o} \frac{1}{\partial_{\tilde{s}} \tilde{R}} \partial_{\tilde{s}} \tilde{B}$$

Using the above expression for $\vec{\nabla} \times \vec{B}$ and the definition of ϵ_2 given by equation (2.30):

$$H_{jul} = \frac{N_o k_B T_o v_o \epsilon_2}{L_o} \left[\frac{1}{\partial_{\tilde{s}} \tilde{R}} \partial_{\tilde{s}} \tilde{B} \right]^2 \qquad (2.46)$$

The various transport coefficients that appear in equation (2.4) are converted to dimensionless values through the following definitions:

$$\tilde{k}_{\perp}^{\epsilon} = k_{\perp}^{\epsilon} \left(\frac{L_o}{k_B N_o v_o R_o^2} \right)$$

$$\tilde{k}_{11}^{\epsilon} = k_{11}^{\epsilon} \left(\frac{1}{k_B N_o v_o L_o} \right) \qquad (2.47)$$

$$\tilde{\tau}_{eq} = \tau_{eq} \left(\frac{v_o}{L_o} \right)$$

$$\tilde{\epsilon}_L = \epsilon_L \left(\frac{L_o}{k_B T_o v_o} \right)$$

If equation (2.4) is now converted to dimensionless magnetic flux coordinates with the aid of equations (2.44) to (2.47) and (2.31) the following expression is obtained.

$$N(\partial_t + v\partial_x + \epsilon_2 D\partial_s) T_\epsilon \; - \; (\gamma-1) T_\epsilon (\partial_t + v\partial_x + \epsilon_2 D\partial_s) N =$$

$$\frac{N(T_{\epsilon'} - T_\epsilon)}{\tau_{eq}} + (\gamma-1) N \epsilon_L + (\gamma-1) \epsilon_2 \left[\frac{1}{\partial_s R} \partial_s B \right]^2 \qquad (2.48)$$

(continued on next page)

$$+ (\gamma-1) \frac{1}{R\partial_s R} \partial_x[R\partial_s R k_{11}^{\epsilon} \partial_x T_\epsilon] + (\gamma-1) \frac{1}{R\partial_s R} \partial_s[\frac{R}{\partial_s R} k_{\perp}^{\epsilon} \partial_s T_\epsilon]$$

where all variables are assumed to be dimensionless and the $\sim$'s have been dropped for clarity.

2.3.4 Magnetic Field

Rather than solve equations for the magnetic field as given in Section 2.1.4, it is more convenient to solve for a function closely related to $\tilde{R}(s,x,t)$. The derivation of the appropriate equations will be found in Section 2.4.

2.3.5 Laser Absorption and Propagation

The dimensionless units that are defined in connection with the description of the propagation and absorption of the laser beam are:

$$\epsilon_L = \left(\frac{k_B T_o v_o}{L_o} \right) \tilde{\epsilon}_L$$

$$k_a = \left(\frac{1}{L_o} \right) \tilde{k}_a$$

$$I_1 = (N_o k_B T_o v_o) \tilde{I}_1$$

$$P_L = (N_o k_B T_o A_o v_o) \tilde{P}_L$$

With these definitions, equations (2.8) and (2.10) transform to the following two equations

$$\epsilon_L = \frac{k_a I_1}{N} \qquad (2.49)$$

$$\frac{1}{c} \frac{\partial}{\partial t} P_L + \frac{\partial}{\partial x} P_L = - \bar{k}_a P_1$$

where all variables are assumed to be dimensionless and the ~'s have been dropped.

2.4 Shell Equations

The ion density, N, axial fluid velocity, *v*, electron and ion temperatures, T_e and T_i, and axial magnetic field, B_z, are now approximated as constants between adjacent discrete flux coordinates that have been chosen for the radial grid. Thus the plasma is represented by a finite number of uniform co-axial shells, each with constant magnetic flux as depicted schematically in Figure 2.3. The plasma variables are now functions x, t, and the shell number which may be indicated by an integer subscript, for example, $N_s(x,t)$, $v_s(x,t)$, etc. Shell subscripts will be omitted for the sake of clarity. In this approximation, the resulting difference equations are accurate to second order in the differences between adjacent flux coordinates when the values of the plasma variables are taken to be the values at the shell centers.

The advantages of this scheme are two-fold. First, when radial pressure balance is assumed, it is a simple matter to decouple the coupled magnetohydrodynamic equations in each shell by using the constancy of the area inside the cylindrical wall which confines the plasma. This will be described later in this section. Secondly, the program may be used for quasi-one-dimensional calculations of high temperature devices by using a few shells when accuracy in the radial direction is not essential but where the effects of diffuse profiles and magnetic diffusion need to be considered. It should be noted here that if a coarse numerical grid in the radial direction were used in a model that did not employ a coordinate system in which one of the coordinates was aligned with the magnetic field, numerical diffusion would be a serious problem.

The shell equations or the magnetohydrodynamic equations for each shell will be derived below. Since dimensionless variables are used exclusively in this subsection the ~'s over all variables are dropped. In these equations, ∇Q denotes the difference between the values of a quantity Q evaluated at the upper and lower boundaries of a shell. Values on a boundary can be calculated as linear interpolations or averages of adjacent shell values. The symbol $\langle Q \rangle$ represents the average value of Q inside the shell when Q is not a simple plasma variable.

2.4.1 Continuity Equation

Multiplication of equation (2.34) by two yields:

$$\partial_t (N\ 2R\partial_s R) + \partial_x (v\ N2R\partial_s R) + \partial_s (\epsilon_2 DN2R\partial_s R) = 0 \qquad (2.50)$$

This equation is now integrated over s from s_L to s_u where s_L is defined as the lower boundary of the s'th shell and s_u is the upper boundary of the s'th shell. Since N and v are assumed constant over the integration the first term becomes:

$$\frac{\partial}{\partial t} [N \int_{s_L}^{s_u} 2R\partial_s R ds] = \frac{\partial}{\partial t} [N \int_{s_L}^{s_u} \partial_s (R^2) ds] = \frac{\partial}{\partial t} (NA)$$

where

$$A \equiv \int_{s_L}^{s_u} \partial_s (R^2)\ ds = R^2 \Big|_{s_L}^{s_u}$$

is the area of the shell being integrated over. Similarly the second term in equation (2.50) becomes:

$$\partial_x N\ v[\int_{s_L}^{s_u} 2R\partial_s R\ ds] = \partial_x (N\ A\ v)$$

With the use of equation (2.27) the last term in equation (2.50) can be written as:

$$\partial_s(\epsilon_2 D\ N\ 2R\partial_s R) = \partial_s\left(\frac{\epsilon_2 DN}{B}\right)$$

thus,

$$\int_{s_L}^{s_u} \partial_s(\epsilon_2 DN2R\partial_s R)\ ds = \left(\frac{\epsilon_2 DN}{B}\right)\Big|_{s_L}^{s_u}$$

The following definitions are made.

$$F_s = \left(\frac{\epsilon_2 DN}{B}\right)_{s+1/2}$$

and

$$\Delta F = F_s - F_{s-1}$$

where a subscript s denotes the s'th shell and the subscript (s + 1/2) denotes the boundary between the s'th and the (s+1)'th shell.

The continuity equation can now be expressed as:

$$\frac{\partial}{\partial t}(NA) + \frac{\partial}{\partial x}(NAv) + \Delta F = 0 \qquad (2.51)$$

It is noted that since the integration has been performed the continuity equation has become a function of the area A of each shell rather than s. The third term can be physically interpreted as a particle flux across the magnetic field lines.

2.4.2 Momentum Equation

If equation (2.40) is multiplied by $2R\partial_s R$ and the result added to 2v times equation (2.34), the following equation is obtained:

$$2NR\partial_s R[\partial_t + v\partial_x + \epsilon_2 D\partial_s]v + 2v[\partial_t(NR\partial_s R) + \partial_x(vNR\partial_s R)$$

$$+ \partial_s(\epsilon_2 DNR\partial_s R)] = -2R\partial_s R\ \partial_x(P+q_x) + 2R\partial_x R\partial_s(P+B^2)$$

The above equation can be written as:

$$\frac{\partial}{\partial t}[Nv\frac{\partial R^2}{\partial s}] + \frac{\partial}{\partial x}[Nv^2\frac{\partial R^2}{\partial s}] + \frac{\partial}{\partial s}[\epsilon_2 D\frac{NV}{B}]$$

$$= \frac{\partial R^2}{\partial s}\ \frac{\partial}{\partial x}(P+q_x) + 2(R\partial_z R)\ \partial_s(P+B^2)$$

If the above equation is now integrated over a shell s assuming that N, v, P, and B are constant over the shell it is found that:

$$\frac{\partial}{\partial t}[NAv] + \frac{\partial}{\partial x}[NAv^2] + \Delta(\epsilon_2\frac{DNv}{B}) = -A\frac{\partial}{\partial x}[P+q_x]$$

$$+ <\partial_x(R^2)> \Delta(P+B^2)$$

If the derivatives on the left hand side are expanded and use is made of equation (2.51) the following equation is obtained.

$$\frac{\partial v}{\partial t} = -v\frac{\partial}{\partial x}v - \frac{[\Delta G - v\Delta F]}{NA} - \frac{1}{N}\frac{\partial}{\partial x}(P+q_x) + \frac{<\partial_x R^2>}{NA}\Delta(P+B^2)$$

(2.52)

where

$$G_s \equiv \left(\frac{\varepsilon_2 DNv}{B} \right)_{s+1/2} = v_{s+1/2} F_s \qquad (2.53)$$

2.4.3 <u>Temperature Equation</u>

Equation (2.48) can be expressed as:

$$N^{\gamma}\left(\frac{\partial}{\partial t} + v\frac{\partial}{\partial x} + \varepsilon_2 D \frac{\partial}{\partial s}\right)(TN^{-(\gamma-1)}) = \frac{N(T_{\varepsilon'} - T_{\varepsilon})}{\tau_{eq}} + (\gamma-1)N\,\varepsilon_L$$

$$+ (\gamma-1)\varepsilon_2\left[\frac{1}{\partial_s R}\partial_s B\right]^2 + (\gamma-1)\frac{1}{R\partial_s R}\partial_x[R\partial_s R k_{11}^{\varepsilon}\partial_x T_{\varepsilon}]$$

$$+ (\gamma-1)\frac{1}{R\partial_s R}\partial_s\left[\frac{R}{\partial_s R} k_{\perp}^{\varepsilon}\partial_s T_{\varepsilon}\right]$$

For the remainder of this subsection the right hand side of the above equation will be referred to as RHS. The above equation is multiplied by $2N^{(1-\gamma)}R\partial_s R$ to obtain the following:

$$2N(R\partial_s R)\left[\frac{\partial}{\partial t} + v\frac{\partial}{\partial x} + \varepsilon_2 D\frac{\partial}{\partial s}\right](TN^{1-\gamma}) = 2N^{1-\gamma}\, R\partial_s R \cdot \mathrm{RHS}$$

From the continuity equation:

$$T_{\varepsilon}N^{1-\gamma}[\partial_t(2NR\partial_s R) + \partial_x(2vNR\partial_s R) + \partial_s(2\varepsilon_2 DNR\partial_s R)] = 0$$

By adding the last two equations the following equation can be obtained.

$$\frac{\partial}{\partial t}[\partial_s R^2 T_\varepsilon N^{(2-\gamma)}] + \frac{\partial}{\partial x}[\partial_s R^2 N^{(2-\gamma)} v T_\varepsilon] + \frac{\partial}{\partial s}[\frac{\varepsilon_2 D N^{(2-\gamma)}}{B} T_\varepsilon]$$

$$= \partial_s R^2 N^{(1-\gamma)} \cdot \text{RHS}$$

If the above equation is integrated over a shell s, noting that N, v, T_ε, and B are constants over a shell the following equation is obtained.

$$\frac{\partial}{\partial t}[AN^{(2-\gamma)} T_\varepsilon] + \frac{\partial}{\partial x}[AN^{(2-\gamma)} v T_\varepsilon] + \Delta[\frac{\varepsilon_2 D N^{(2-\gamma)} T_\alpha}{B}]$$

$$= \int_{s_L}^{s_u} \partial_s R^2 N^{(1-\gamma)} \cdot (\text{RHS})\, ds$$

The variable I_s is defined as:

$$I_s = \frac{\varepsilon_2 D N}{B} T_\varepsilon N^{(1-\gamma)} \Big|_{s+1/2} = (N^{(1-\gamma)} T_\varepsilon) \Big|_{s+1/2} F_s \quad (2.54)$$

Now, by expanding derivatives the above equation can be written as:

$$N^{(1-\gamma)} T_\varepsilon [\partial_t (NA) + \partial_x (NAV)] + NA[\partial_t + v\partial_x](N^{(1-\gamma)} T_\varepsilon)$$

$$+ \Delta I = \int_{s_L}^{s_u} \partial_s R^2 N^{(1-\gamma)} (\text{RHS})\, ds$$

A value for the first term in brackets may be obtained from equation (2.51) and is equal to $-\Delta F$. By using this expression and dividing the above equations by $AN^{(1-\gamma)}$ and evaluating RHS, the following equation is obtained:

$$N[\partial_t + v\partial_x]\, T_\varepsilon - (\gamma-1)\, T_\varepsilon[\partial_t + v\partial_x]N + \frac{1}{NA}\,[N^\gamma\, \Delta I - NT_\varepsilon\, \Delta F]$$

$$= \frac{N(T_{\varepsilon'} - T_\varepsilon)}{\tau_{eq}} + (\gamma-1)\, N\, \varepsilon_L + (\gamma-1)\, \varepsilon_2\, [\,\frac{1}{\partial_s R}\ \partial_s B]^2 \qquad (2.55)$$

$$+ \frac{(\gamma-1)}{A}\, \partial_x[A\, k^\varepsilon_{11}\, \partial_x\, T_\varepsilon] + \frac{(\gamma-1)}{A}\ \Delta[\,\frac{2R}{\partial_s R}\, k^\varepsilon_\perp\, \partial_s T_\varepsilon]$$

It is convenient to rewrite this equation as:

$$\partial_t\, T_\varepsilon = \frac{(\gamma-1)\, T_\varepsilon'\, \partial_t\, N}{N} + H^+_\varepsilon + \frac{(\gamma-1)}{NA}\, \Delta[2R\, k^\varepsilon\, \frac{1}{\partial_s R}\, \partial_s\, T_\varepsilon]$$

$$+ \frac{(\gamma-1)}{NA}\ \partial_x[A\, k^\varepsilon_{11}\, \partial_x\, T_\varepsilon] \qquad (2.56)$$

where

$$H^+_\varepsilon = -\, v\, \partial_x\, T_\varepsilon + (\gamma-1)\, T_\varepsilon\, v\, \frac{\partial_x N}{N} - \frac{1}{NA}\ [N^{(\gamma-1)}\, \Delta I - T_\varepsilon\, \Delta F] + (\gamma-1)\, \varepsilon_L$$

$$+ \frac{(T_{\varepsilon'} - T_\varepsilon)}{\tau_{eq}} + \frac{(\gamma-1)\, \varepsilon_2}{N}\ (\,\frac{1}{\partial_s R}\, \partial_s B)^2 \qquad (2.57)$$

The third term on the right hand side of equation (2.56) represents diffusion of heat across the magnetic field while the fourth term represents heat flow along the field lines.

2.4.4 <u>Closure of the System of Equations</u>

Equations (2.51), (2.52), and (2.56) describe four dynamic equations for five plasma variables, namely N, v, T_e, T_i, and A. To complete the set of equations, an equation for A must be derived from the equations describing the radial dynamics of the plasma. That is, an equation for $\partial A/\partial t$ is required. If the time scale over which effects occur is much longer than that required for a magnetosonic wave to travel across the plasma, then the assumption that pressure balance exists is valid. In this case, any pressure imbalance is communicated sufficiently rapidly across the plasma column so as to insure that pressure balance is essentially being maintained. This assumption would apply to physical situations like that of a long reactor plasma. In the case where a plasma is rapidly heated by a laser, causing both radial and axial shocks, full radial dynamics must be included. The computer program has been developed so that two options exist, namely that of using full radial dynamics or that of pressure balance. The assumption of radial pressure balance, where it is valid, leads to considerable savings in CPU time.

2.4.4a Radial Pressure Balance

In this subsection radial pressure balance is assumed in the derivation for an expression for $\partial A/\partial t$. For this case equation (2.38) for v_r is not used. The assumption of radial pressure balance is expressed as:

$$\frac{\partial}{\partial s} (P + B^2) = 0 \quad \text{or} \quad P + B^2 = \pi(x,t) \qquad (2.58)$$

where $\pi(x,t)$ is the total kinetic plus magnetic pressure and is not a function of s. If ϕ is the magnetic flux in a shell, then $B = \phi/A$ where ϕ is a constant in each shell. By using this expression for B in equation (2.58), and differentiating with respect to time yields:

$$\partial_t P - 2\frac{\phi^2}{A^3}\partial_t A = \partial_t \pi(x,t)$$

or

$$A(\partial_t P) = 2\, B^2(\partial_t A) + A(\partial_t \pi) \qquad (2.59)$$

Now the temperature equation can be expressed as:

$$AN(\partial_t T_\varepsilon) = (\gamma-1)\; T_\varepsilon\; A(\partial_t N) + H_\varepsilon$$

where

$$H_\varepsilon \equiv H_\varepsilon^+ + \frac{(\gamma-1)}{NA}\,\Delta[2R\; k_\perp^\varepsilon \frac{1}{\partial_s R}\,\partial_s T_\varepsilon] + \frac{(\gamma-1)}{NA}\,\partial_x[A\; k_{11}^\varepsilon\; \partial_x T_\varepsilon]$$

Define,

$$H \equiv H_e + H_i \quad \text{and} \quad T = T_e + T_i$$

and add the two temperature equation to obtain:

$$AN(\partial_t T) = (\gamma-1)\; T\; A(\partial_t N) + H$$

Now,

$$\partial_t(P) = \partial_t(NT) = N\,\partial_t T + T\,\partial_t N$$

Combining this with the above equation yields:

$$A(\partial_t P) - \gamma TA(\partial_t N) = H \tag{2.60}$$

The continuity equation is used to define C as:

$$C \equiv \partial_t (NA) = - [\partial_x (NAv) + \Delta F]$$

or

$$\partial_t N = [C - N \partial_t A]/A$$

Substituting this into equation (2.60) yields:

$$A(\partial_t P) = H + \gamma T[C - N(\partial_t A)] \tag{2.61}$$

Comparing equations (2.59) and (2.61) yields an expression for $\partial_t A$.

$$(\partial_t A) = [\frac{H + \gamma TC}{2B^2 + \gamma P}] - \frac{A \partial_t \pi}{2B^2 + \gamma P} \tag{2.62}$$

Before equation (2.62) can be used an expression for $\partial_t \pi$ must be derived. This may be done by noting that the total cross sectional area of the solenoid is a constant. If equation (2.62) is summed over all shells.

$$\sum_{\text{All shells}} (\partial_t A) = 0$$

Thus,

$$(\partial_t \pi) = \sum_{\text{All Shells}} \frac{H + \gamma TC}{2B^2 + \gamma P} \bigg/ \sum_{\text{All Shells}} \frac{A}{2B^2 + \gamma P} \qquad (2.63)$$

2.4.4b Full Radial Dynamics

In this sub-section an expression for $\frac{\partial A}{\partial t}$ is derived by using the appropriate equations to incorporate full radial dynamics into the model. First an expression for $\frac{\partial}{\partial t}(R^2)$ is found. It is then a trivial matter to find an expression for $\frac{\partial A}{\partial t}$. In order to obtain an equation for R^2, equation (2.19) is multiplied by 2R. The resulting equation can be written as:

$$\partial_t(R^2) + v\, \partial_x(R^2) + 2R\, \varepsilon_2\, D\, \partial_s R = 2Rv_r$$

The third term in the above equation can be reduced by expanding D through the use of equations (2.29) and (2.27).

$$D = R\frac{\partial B}{\partial r} = \frac{R}{\partial_s R}\partial_s B = \frac{R}{\partial_s R}\partial_s\left(\frac{1}{2R\partial_s R}\right) = \frac{R}{\partial_s R}\partial_s\left(\frac{1}{\partial_s R^2}\right)$$

The third term therefore becomes:

$$2R^2\, \varepsilon_2\, \partial_s\left(\frac{1}{\partial_s R^2}\right) = \frac{-2R^2\, \varepsilon_2\, \partial_{ss}(R^2)}{(2R\, \partial_s R)^2} = -\, 2R^2B^2\, \varepsilon_2\, \partial_{ss}(R^2)$$

The equation for R^2 can now be written as:

$$\frac{\partial}{\partial t}(R^2) = -\, v\frac{\partial}{\partial x}(R^2) + 2Rv_r + 2\,\varepsilon_2\, R^2\, B^2\, \partial_{ss}(R^2) \qquad (2.64)$$

The solution to equation (2.64) requires an expression for Rv_r. An equation for Rv_r can be found as follows. Note that

$$\frac{dR}{dt} = v_r$$

Thus,

$$N \frac{d}{dt} (Rv_r) = Nv_r^2 + NR \frac{dv_r}{dt}$$

An expression for $\frac{dv_r}{dt}$ is given by equation (2.38) so that the above expression can be written as:

$$N[\partial_t + v\partial_x + \epsilon_2 D\partial_s] (Rv_r) = \frac{N(Rv_r)^2}{R^2} - \frac{2B(\text{[illegible]}}{\epsilon_1}\partial_s(P + B^2 + q_r)$$

(2.65)

An expression for D is required for equation (2.65). By using equations (2.29) and (2.27), D can be written as:

$$D = R \frac{\partial B}{\partial r} = \frac{R}{\partial_s R} \partial_s B = \frac{2R^2}{2R\partial_s R} \partial_s B = 2R^2 B \partial_s B . \qquad (2.66)$$

Thus equation (2.65) can be used to find Rv_r, and equation (2.64) can be used to find $R^2(s,x,t)$. In a shell model $\frac{\partial}{\partial t} (R^2)_{s+1/2}$ is found where $(R^2)_{s+1/2}$ is the outer radius of a shell. Then the desired quantity $\frac{\partial A}{\partial t}$ can be found.

$$\frac{\partial}{\partial t} A = \frac{\partial}{\partial t} (R^2)_{s+1/2} - \frac{\partial}{\partial t} (R^2)_{s-1/2} \qquad (2.67)$$

2.4.5 <u>Summary of Shell Equations</u>

In this sub-section the shell equations are summarised and written in a form suggestive of how they might be solved through numerical techniques. In this sub-section the notation $\dot{Q}$ and Q' will be used for $\partial_t Q$ and $\partial_x Q$ respectively. The first step is a solution for the area, A_1, of a shell.

If radial pressure balance is assumed equation (2.62) is used.

$$\dot{A} = \frac{H + \gamma TC - A\,\dot{\pi}(x,t)}{2B^2 + \gamma P} \qquad (2.68)$$

where

$$\dot{\pi} = \left[\sum_{\substack{\text{All}\\ \text{Shells}}} \frac{H + \gamma TC}{2B^2 + \gamma P}\right] \Bigg/ \left[\sum_{\substack{\text{All}\\ \text{Shells}}} \frac{A}{2B^2 + \gamma P}\right] \qquad (2.69)$$

and $C \equiv -(NAv)' - \Delta F$ (2.70)

If full radial dynamics are used an equation for A comes from $\dot{W}$ defined on shell boundaries where

$$W \equiv R^2 \qquad (2.71)$$

From equation (2.67)

$$\dot{A} = \dot{W}_{s+1/2} - \dot{W}_{s-1/2} \tag{2.72}$$

and from equation (2.64)

$$\dot{W} = - v W' + 2U + 2\varepsilon_2 W B^2 \frac{\partial^2}{\partial s^2} W \tag{2.73}$$

where

$$U \equiv R v_r \tag{2.74}$$

U can be found through a solution to equation (2.65) which can be expressed as:

$$\dot{U} = - v U' - \varepsilon_2 D \partial_s U + \frac{U^2}{W} - \frac{2BW}{\varepsilon_1^2 N} \partial_s [P + B \text{ [illegible] }_r] \tag{2.75}$$

Once $\dot{A}$ has be found, $\dot{N}$ can be found through solution of the following equation, which comes from equations (2.51) and (2.70):

$$\dot{N} = \frac{(C - N\dot{A})}{A} \tag{2.76}$$

Now the temperature equations can be solved. With the new notation equations (2.56) and (2.57) can be written as:

$$\dot{T}_\varepsilon = H_\varepsilon^* + \theta_{11} + (\gamma-1) T_\varepsilon \dot{N}/N \tag{2.77}$$

where

$$H^*_\epsilon \equiv - v\, T'_\epsilon + (\gamma-1)\, T_\epsilon\, v\, N'/N - \frac{1}{NA}\, [N^{(\gamma-1)}\, \Delta I - T_\epsilon\, \Delta F]$$

(2.78)

$$+ (\gamma-1)\, \epsilon_L + \frac{(T_{\epsilon'} - T_\epsilon)}{\tau_{eq}} + \frac{(\gamma-1)\epsilon_2}{N}\, (\frac{1}{\partial_s R}\, \partial_s B)^2$$

$$+ \frac{(\gamma-1)}{NA}\, \Delta[2R\, k^\epsilon_\perp\, \frac{1}{\partial_s R}\, \partial_s\, T_\epsilon]$$

and

$$\theta_{11} = \frac{(\gamma-1)}{NA}\, [A\, k^\epsilon_{11}\, T'_\epsilon]' \qquad (2.79)$$

Finally, the axial velocity can be found. Equation (2.52) can be expressed as:

$$\dot{v} = - v\, v' - \frac{1}{NA}\, [\Delta G - v\, \Delta F] - \frac{1}{N}\, [P + q_x]' + \frac{\langle W' \rangle}{NA}\, \Delta(P + B^2)$$

(2.80)

This set of equations requires the evaluation of F_s, I_s, and G_s on shell boundaries. From equation (2.66):

$$\frac{D}{B} = 2R^2\, \partial_s B$$

Substituting this into the definition of F_s:

$$F_s = 2R^2\, \epsilon_2 N\, \partial_s B \qquad (2.81)$$

G_s and I_s can be evaluated from:

$$G_s = v_{s+1/2} F_s \qquad (2.82)$$

$$\text{[illegible]}\, N^{(1-\gamma)} T_\varepsilon) \Big|_{s+1/2} F_s$$

A computer model of a plasma in a solenoidal magnetic field can be made through the numerical solution of the coupled equations presented in this sub-section.

2.5 Numerical Solution

A computer simulation based on the shell equations of the previous sub-section has been developed. The main computer routine of this simulation solves the appropriate difference equations which have been written in dimensionless units. As the solution is advanced in time the internal variables of the routine are output on magnetic tape or disk. (This is essentially the only output of the main routine. Subsequently, post-processor routines read the output of the main routine and calculate values for various plasma parameters in a more familiar unit system. Different post-processor routines have been developed to print out values for plasma parameters and perform checks on the validity of the simulation. Two checks that are done to test the validity of a simulation are conservation of energy and conservation of mass. These will be discussed later in this section. In addition to these post-processor routines two graphics routines have been developed. One of these routines plots various two-dimensional plots of plasma parameters as functions of x, r, or t. The other makes three dimensional plots of various plasma parameters as functions of x and r at given times. Examples of output from these routines will be presented in chapters three and four.

An attempt has been made to make a package of programs that will allow a user with a minimal familiarity with the programming details to solve a variety of problems with little effort. Various options have been built into the routines and desired options are specified in an input file. A more detailed description of the routines and some details of the numerical techniques can be found in Appendix A.

The shell equations have been solved using a two-step second order Euler method. In the first step, the solution at t is used to calculate temporary values of the variables at $t + \Delta t/2$. These temporary values are used to calculate the coefficients and spatial derivatives in the equations and the solution is then advanced from t to $t + \Delta t$ in the second step.

Central differences are used to calculate all spatial derivatives. The continuity and axial acceleration equations are advanced explicitly. When radial pressure balance is assumed, the equation for $\frac{\partial A}{\partial t}$ is also advanced explicitly. However the decoupling procedure used to find $\frac{\partial A}{\partial t}$ is similar to an implicit scheme in the sense that changes in the central pressure due to heating are felt instantly in all the shells. When full radial dynamics are taken into account, the radial acceleration equation is solved explicitly but the shell boundary equation for W, equation (2.73), is solved implicitly. This is necessary since large magnetic diffusion in regions of low electron temperature severely limits the timestep of an explicit scheme.

For each timestep, the temperature equations are advanced in two stages. In the first step, parallel heat conduction is turned off and the equations are advanced using implicit differences for the perpendicular conduction terms. The difference between the new and old values of T_ε is taken to be the contribution to $\frac{\partial T_\varepsilon}{\partial t} \Delta t$ from perpendicular conduction. This quantity is substituted for the perpendicular terms in the second stage in which the equations are advanced using implicit differences for the parallel conduction terms.

The boundary conditions used at the outer wall are $V_r = 0$, W = constant, and T_ε = constant or $\frac{\partial T_\varepsilon}{\partial s} = 0$. At the open ends, the plasma variables are assumed to vary linearly with x. The exception to this rule is that $\frac{\partial P}{\partial x}$ is set equal to a constant in order to get the flow started at $t = 0$. The results are insensitive to the particular value of $\frac{\partial P}{\partial x}$ chosen.

The code was tested in several ways. Axial flow without heat conduction was compared to analytic similarity solutions from a one-shell model[18]. Axial heat conduction was tested by freezing the plasma ($v=0$, $N=N_o$) and assuming a constant conduction coefficent for which an analytic solution is easily obtainable from reasonable initial conditions. In both cases, excellent agreement was obtained. The full radial dynamics under axially uniform laser heating was compared with results from a one-dimensional code described by Burnett and Offenberger[37]. Essentially identical results were obtained.

A check on conservation of energy and mass is made for each simulation. This is done by calculating total internal energy or mass of the plasma at various times and comparing changes in internal energy or mass with the calculated energy or mass flow through the solenoid ends. Typically the simulations conserve energy to better than 5% and mass to better than 1%.

A more detailed description of numerical techniques are presented in Appendix A.

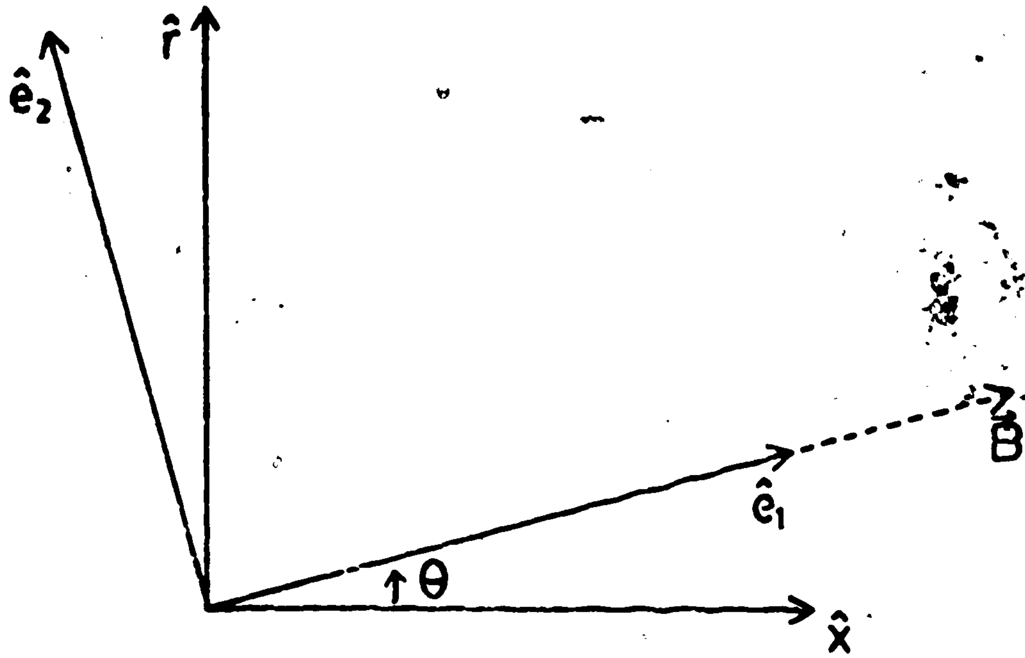

Figure 2.1 Orthogonal coordinate system aligned with magnetic field.

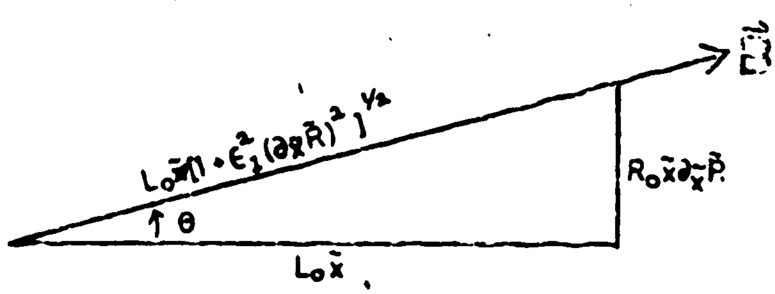

Figure 2.2 Angle of magnetic field lines relative to solenoid axis in magnetic flux coordinate system

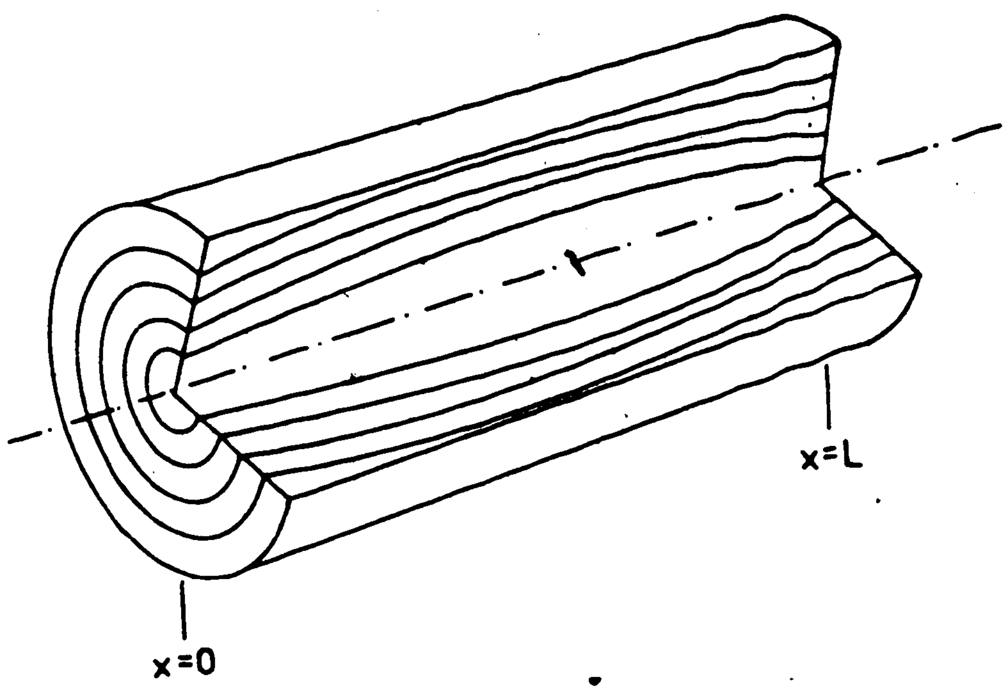

Figure 2.3 Cutaway of plasma column showing shell structure

CHAPTER 3

Computer Simulations of Plasma in a Solenoidal Magnetic Field

In this chapter some results from the computer program described in the previous chapter are presented. The versatility of the code is demonstrated through the simulation of short solenoids (5 cm length) in Section 3.1, intermediate length solenoids (1 m length) in Section 3.2, and long solenoids (1 km length) in Section 3.3. A comparison of a simulation in which full radial dynamics have been included to a simulation that assumes radial pressure balance is presented in Section 3.1. Radial pressure balance is assumed for simulations in Section 3.2 and Section 3.3. The parameters used for the simulations in Section 3.2 were chosen to test the feasibility of creating plasma conditions ideally suited for the beat frequency mixing of antiparallel laser beams in a laser heated solenoid. The required plasma conditions will be discussed in Chapter 6 where the results from Section 3.2 will be referred to. Results of Section 3.3 demonstrate the ability of this type of code to simulate the magnetohydrodynamic behaviour of a 1 km solenoid reactor.

3.1 Laser Heated Solenoid

In the first simulation, a preionised hydrogen plasma of radius 1.5 cm is heated by a CO_2 laser with Gaussian cross-section and half-power radius 1.77 mm. The laser power rises linearly from 0 at t=0 to 100 MW at t=10 nsec and then remains constant. The initial plasma variables are:

$$n_o = 2 \times 10^{18} \text{ cm}^{-3}$$

$$T_e = T_i = 1 \text{ eV}$$

$$B = 100 \text{ kgauss}$$

The parameters were chosen to be the same as those used in an experimental and theoretical study by Scudder *et al.*[16] with the important difference that their plasma was formed by the breakdown of neutral gas by the laser.

The code was run with 60 axial points and 30 shells until t=0.5 μsec was reached. This run required 20 minutes of CPU time on an Amdahl 470 V/6. The size of each timestep was limited by the hydrodynamic stability condition,

$$\Delta t < \Delta r / V_{ma}$$

where Δr is the shell thickness and V_{ma} is the magneto-acoustic velocity. Three dimensional plots of T_e, T_i, n_o, v_r, v_x and B_x are presented in Figures 3.1 to 3.6. In each plot, the lines joining the plotted values of the plasma variables are drawn at a constant axial or radial position so that linear interpolation between shell values was necessary.

The electron temperature attained, 40-45 eV, is similar to those calculated by Scudder et al.[16] in a two-dimensional Lagrangian simulation. The ion temperature distribution is almost identical to that of the electrons since the collision time is short compared to the heating time at these densities. The temperature plateau behind the bleaching front is apparently due to adiabatic cooling which follows expansion of the heated plasma.

The most significant difference between this simulation and that of Scudder et al. is the nature of the plasma dynamics at the bleaching front. They have found that after t=140 nsec the laser driven shock on the axis attenuates and has disappeared at t=280 nsec. Figure 3.3 indicates, however, the presence of a strong shock at the bleaching front at t=500 nsec. (The jagged edge on the curved shock front is a plotting routine effect.) The shock wave calculated by the shell code has been found to propagate more as a travelling wave rather than a damped wave. The discrepancy between the two codes may be due to the different methods of calculation of the laser beam propagation. Scudder et al. assumed that the propagation was parallel to the axis so that the rays in the outer edges are always in a region of higher absorption than those in the center. The effect was that the beam became progressively narrower as it propagated through the plasma. Eventually, the heated region at the bleaching front may have been too slender to drive a forward moving shock wave. On the other hand, since the shell code assumes a constant beam radius, the heated region always has the same width. To get a more accurate picture of the dynamics at the bleaching front, a better laser model will have to be developed.

The radial velocity distribution is plotted in Figure 3.4. The interesting points are that (1) the velocity is small compared to the magneto-acoustic velocity so that not much energy goes into radial motion and (2) the radial velocity quickly becomes negligible after the passage of the region of fast heating indicating the return to radial equilibrium. These observations suggest that similar densities and temperatures will be obtained if the plasma is assumed to be in radial pressure balance at all times.

The computer simulation was repeated with all of the same input parameters as above, except that radial pressure balance was assumed and the number of shells was reduced to 15. Three dimensional photos of T_e, T_i, n_σ, v_z and B_z are presented in Figures 3.7 to 3.11. The temperatures attained are very similar in the two runs. The densities are also similar, especially in the quiescent region between x = 0 and x=3 cm. The main differences with pressure balance are that (1) the bleaching wave propagates about 10% farther in 500 nsec and (2) the amplitude of the shock is smaller.

The advantage of assuming pressure balance is the significant reduction of computing costs. The run with pressure balance required about 6 minutes of CPU time compared to 20 minutes for full radial dynamics. This reduction of computing time is possible because the timestep is not restricted by the hydrodynamic stability condition when radial pressure balance is assumed. The criterion used for limiting the timestep in this run was that the temperature should not rise by more than 30% at any point in a single step. The assumption of radial pressure balance leads to even greater savings in modelling longer devices and when a laser beam bleaching front is not pushing a shock wave.

3.2 Production of Long Scale Length Plasma for Beat Frequency Mixing

In Chapter 5 a theory for the beat-frequency mixing of anti-parallel laser beams in a homogeneous plasma will be presented. In Chapter 6 the possibility of using this process to create a useful amplifier of infrared radiation will be examined. The possibility of making such an amplifier is dependent on the feasibility of creating a hot plasma with a uniform density and temperature in the axial direction. In this section computer simulations are used to examine the feasibility of creating such a plasma through the inverse bremsstrahlung absorption of laser radiation in a plasma radially confined by a solenoidal magnetic field.

Results from two simulations are presented. The input parameters for both simulations are identical except for initial plasma densities. In both cases a preionized hydrogen plasma is assumed to be contained by a solenoid of radius 1.5 cm, length 1 m, and is heated by a CO_2 laser with Gaussian cross-section and half-power radius of 1.77 mm. The laser power rises linearly from 0 at t=0 to 1 GW at t=10 ns and then remains constant. Initially the temperature, density, and magnetic field are uniform. The temperature is assumed to be 1 eV and the magnetic field strength is assumed to be 100 kG. The computational grid consists of 30 axial points and 15 shells. Under these conditions the code requires approximately 1 minute of CPU time on the Amdahl 470 V/6 computer to simulate the plasma for 1 μsec.

In the first simulation the initial plasma density was 7.8×10^{16} cm^{-3}. Figures 3.12 to 3.14 are three dimensional plots of T_e, T_i, and n_o at t=0.5 μsec. Figure 3.15 is a plot of the axial values of T_e, T_i and n_o as a function of x. These figures show that the axial values of the plasma temperature and density show little variation down most of the column length. For about 80% of the solenoid length the plasma density is almost exactly 7.5×10^{16} cm^{-3} while electron temperature varies from about 95 eV to 100 eV and the ion temperature varies from about 67 eV to 71 eV. It will be shown in Chapter 6 that such plasma conditions are ideally suited to the beat frequency mixing of antiparallel electromagnetic beams with wavelengths of 9.6 μm and 10.6 μm.

The uniform plasma density and temperatures in the axial direction can be attributed to two main factors. First, the high laser powers and low initial plasma densities lead to high laser bleaching velocities. In the above simulation the laser has bleached its way through the plasma column in less than 40 ns. This leads to a relatively uniform laser heating rate down the column axis. The second factor leading to uniform axial profiles is the large electron thermal conductivity in the axial direction which tends to smooth out any nonuniformities.

In the second simulation the initial plasma density was chosen to be 3×10^{17} cm^{-3}. This is a much higher initial density than that of the previous run. Consequently a much lower bleaching velocity is expected. It was found that it took about 300 ns for the laser beam to bleach its way through the solenoid. Because of the slower bleaching velocity, the plasma must be heated for a longer period of time for its axial parameters to become uniform. Figures 3.16 to 3.18 are three dimensional plots of T_e, T_i, and n_o after 1 μsec of laser heating. Figure 3.19 is a plot of the axial values of T_e, T_i, and n_o as a function of x. These figures show that the plasma has a density of between 2.4×10^{17} and 2.5×10^{17} cm^{-3}, an electron temperature of between 120 and 180 eV and an ion temperature of between 100 and 140 eV over an 80 cm length of the 1 meter solenoid. While these parameters are not as uniform as those in the previous simulation it will be shown in Chapter 6 that this plasma would serve very well as the mixing medium for antiparallel radiations with difference frequencies close to the ion-acoustic frequency.

The results of these simulations suggest that a laser heated plasma, radially confined by a solenoidal magnetic field, will have a uniform axial density and temperature profile at a time t after the laser is turned on if:

$$\frac{L_o}{v_{bl}} << t << \frac{L_o}{v_{ac}}$$

where L_o is the solenoid length, v_{bl} is the bleaching wave velocity and v_{ac} is the acoustic velocity. The relation $L_o/v_{bl} << t$ allows laser energy to be deposited uniformly along the solenoid axis while the relation $t << L_o/v_{ac}$ ensures that the effects of end loss are experienced only near to the solenoid ends. The above relation can obviously be satisfied only if the laser intensity and plasma density are such that $v_{bl} >> v_{ac}$.

3.3 θ-Pinch Reactor

The magnetodydrodynamic behaviour of a 1 km θ-pinch reactor in radial equilibrium was simulated with a 10 shell model and 60 axial points. The initial density and temperature profiles were taken to be:

$$n_o(r,x) = \frac{4\times10^{17}\ cm^{-3}}{1 + \exp((r-1.5)/0.15)}$$

and

$$T_{e,i}(r,x) = \frac{5000\ eV}{1 + \exp((r-2.0)/0.10)}$$

The external magnetic field was set at 405 kguass corresponding to a value of $\beta = 0.98$ on the axis and the magnetic field in each shell was calculated assuming radial pressure balance. Symmetric boundary conditions around x=500 m were assumed. There was no external heating and α particle effects were not taken into account.

The results for two times (0.5 ms and 2 ms) are presented in Figures 3.20 to 3.29. Because symmetry is assumed around the point x=500 m, MHD quantities are only plotted in the region defined by $0 \leq x \leq 500$ m. Figures 3.20 and 3.25 show the radius of the shell boundaries used in the numerical calculations as a function of x. These shell boundaries can also be thought of as magnetic flux lines. In the remaining figures three dimensional plots of T_e, n_o, v_x, and B_x are presented. The ion temperature is almost identical to that of the electrons and has not been plotted. At t=0.5 ms, the effects of self-mirroring are clearly evident in Figure 3.20. Figures 3.22 and 3.23 for the plasma density and velocity show that the greatest loss of shell mass occurs between 1 and 2 cm off-axis. This is in agreement with a theoretical one shell model[17] which predicts that the mass flux at the end is proportional to $(1-\beta)^{1/2} T_e^{1/2}$. This function reaches a maximum for r > 0 since β and T_e decrease off-axis. It is also observed from Figure 3.23 that the effects of the open ends propagate faster in the region of lower β as expected from one-dimensional theory.

At t=2 ms, the effects of the open solenoid ends have reached the midplane in all shells and 72% of the mass has flowed from the end of the reactor. Figure 3.25 shows how the shell boundaries, or magnetic flux lines, have moved in towards the solenoid center as the plasma flows out the ends. Figure 3.26 shows that adiabatic expansion has led to a decrease in plasma temperature from 5000 eV to about 3200 eV. Figure 3.27 shows that while 72% of the mass has flowed from the reactor ends, the on axis density has only decreased by about 12%. This is because, as the mass flows from central portions of the reactor, the magnetic field compresses the plasma radially towards the center. This effect could be important since the thermonuclear reaction rate is proportional to n_o^2. The velocity distribution function, Figure 3.28, as a function of x is found to be very nearly linear in all shells as suggested in the one shell theory of McMullin and Capjack[18]. Figures 3.24 and 3.29 illustrate how the magnetic field fills the central portion of the solenoid as the plasma flows out.

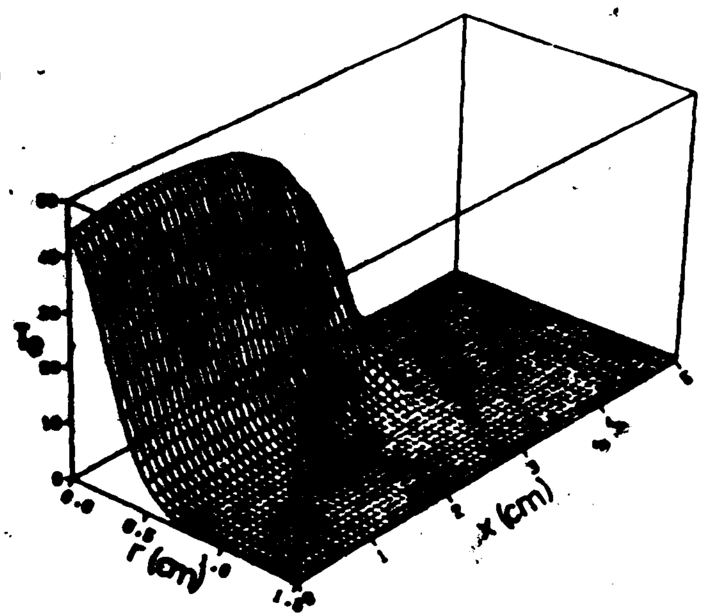

Figure 3.1 Electron temperature in eV at 0.5 μsec; obtained from the first simulation reported in Section 3.1. (Full radial dynamics).

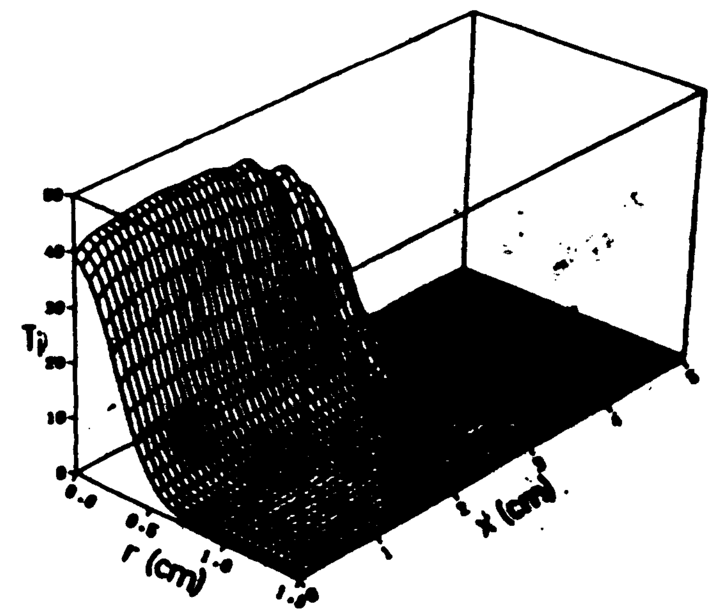

Figure 3.2 Ion temperature in eV at 0.5 μsec obtained from the first simulation reported in Section 3.1. (Full radial dynamics).

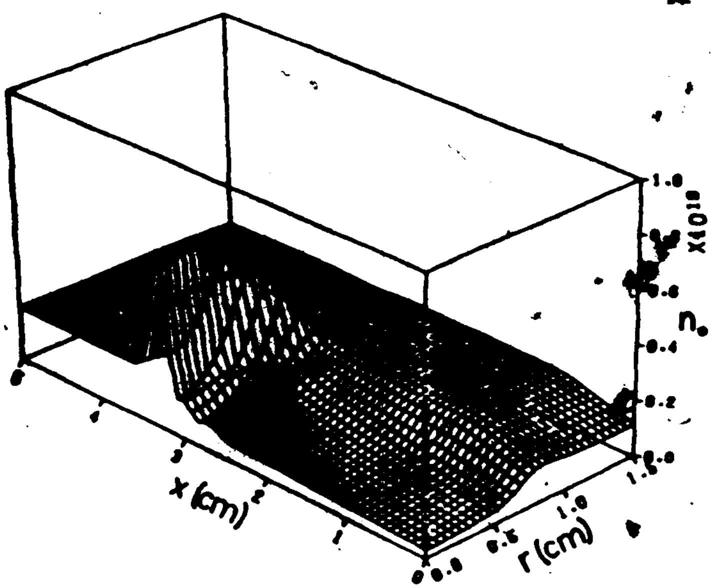

Figure 3.3 Plasma density in cm^{-3} at 0.5 μsec; obtained from the first simulation reported in Section 3.1. (Full radial dynamics).

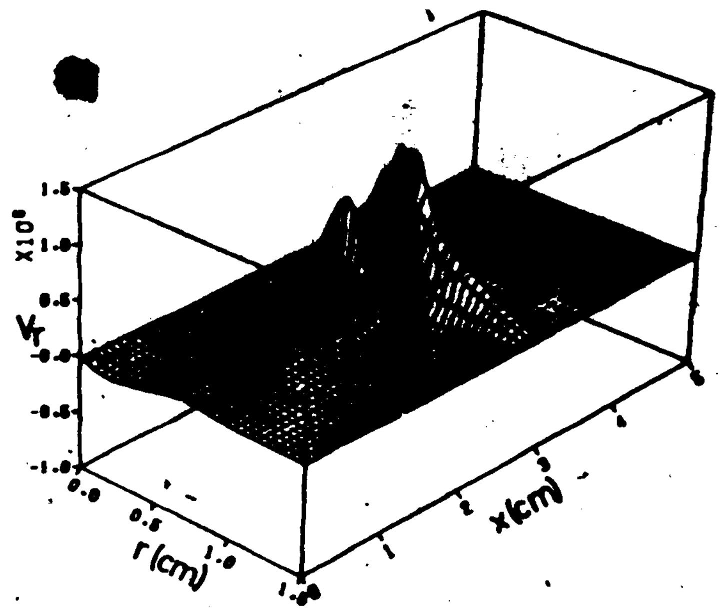

Figure 3.4 Radial plasma velocity in cm/sec at 0.5 μsec; obtained from the first simulation reported in Section 3.1. (Full radial dynamics).

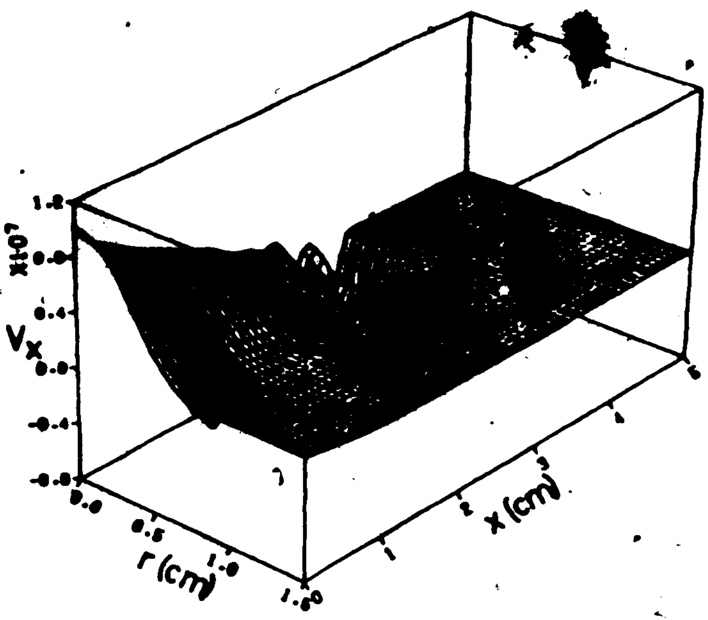

Figure 3.5 Axial plasma velocity in cm/sec at 0.5 μsec; obtained from the first simulation reported in Section 3.1. (Full radial dynamics).

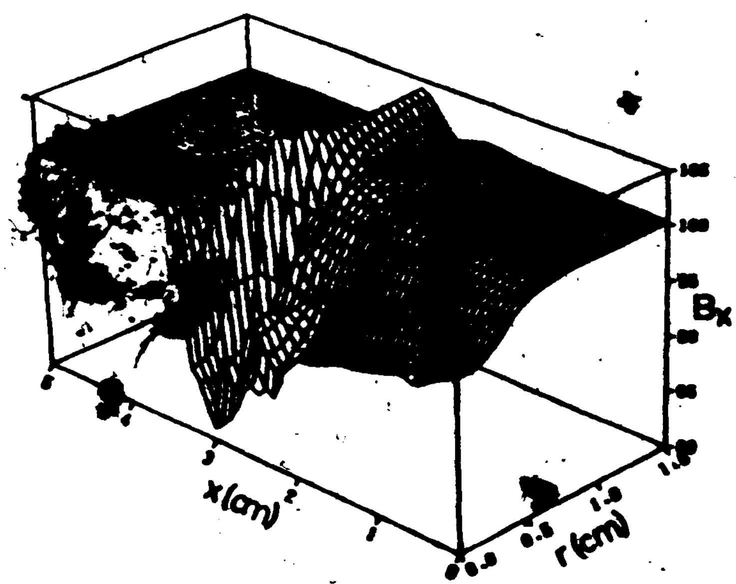

Figure 3.6 Axial magnetic field strength in kG at 0.5 μsec; obtained from the first simulation reported in Section 3.1. (Full radial dynamics).

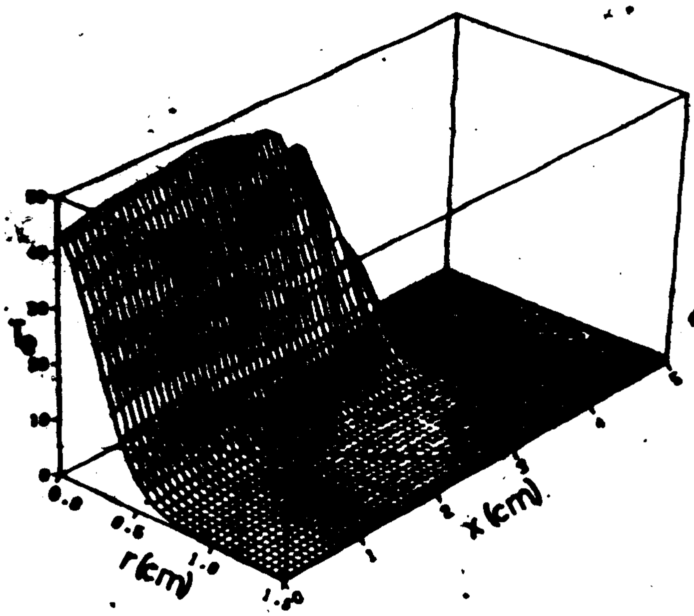

Figure 3.7 Electron temperature in eV at 0.5 μsec; obtained from the second simulation reported in Section 3.1. (Radial pressure balance).

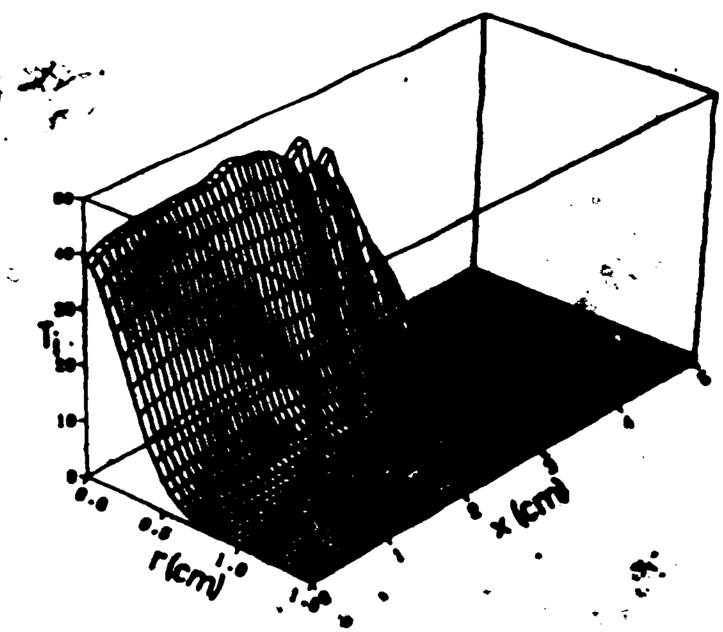

Figure 3.8 Ion temperature in eV at 0.5 μsec; obtained from the second simulation reported in Section 3.1. (Radial pressure balance).

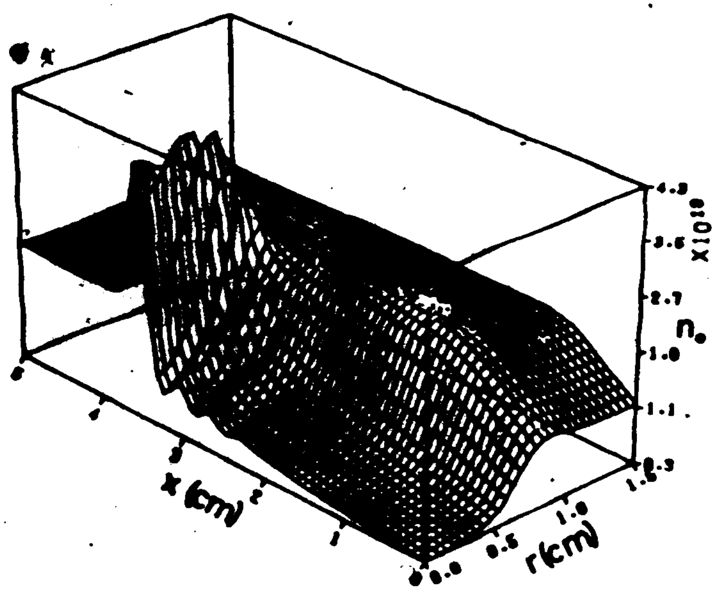

Figure 3.9 Plasma density in cm^{-3} at 0.5 μsec; obtained from the second simulation reported in Section 3.1. (Radial pressure balance).

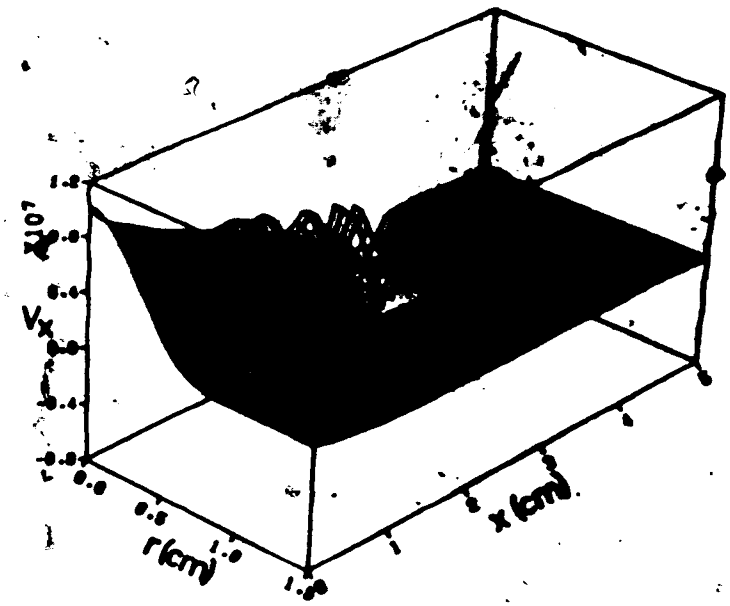

Figure 3.10 Axial plasma velocity at 0.5 μsec; obtained from the second simulation reported in Section 3.1. (Radial pressure balance).

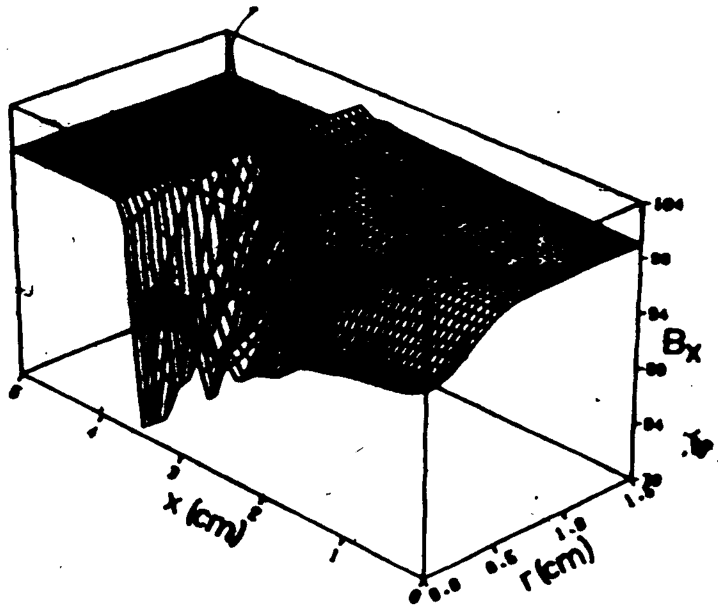

Figure 3.11 Axial magnetic field strength at 0.5 μsec; obtained from the second simulation reported in Section 3.1. (Radial pressure balance).

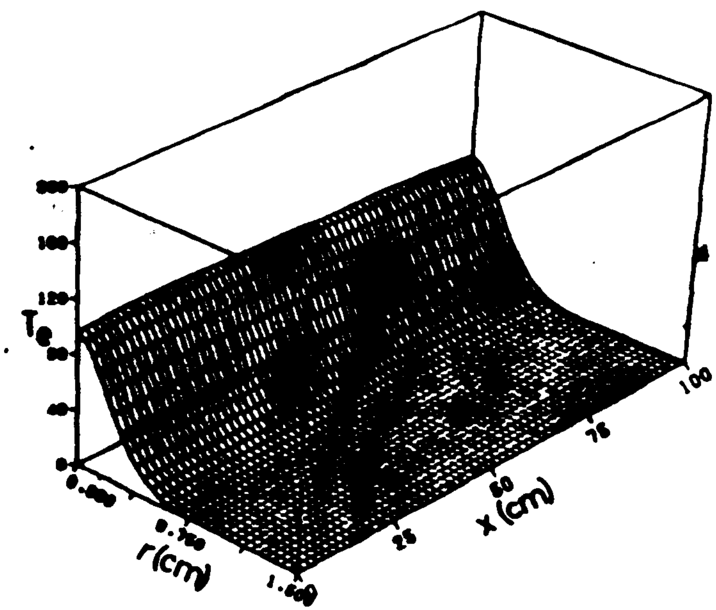

Figure 3.12 Electron temperature in eV at ~0.5 μsec; obtained from the first simulation reported in Section 3.2.

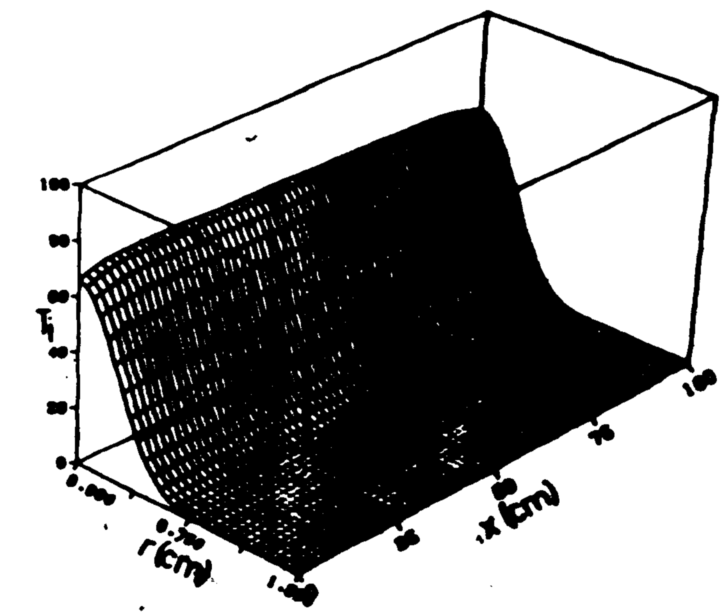

Figure 3.13 Ion temperature in eV at 0.5 μsec; obtained from the first simulation reported in Section 3.2.

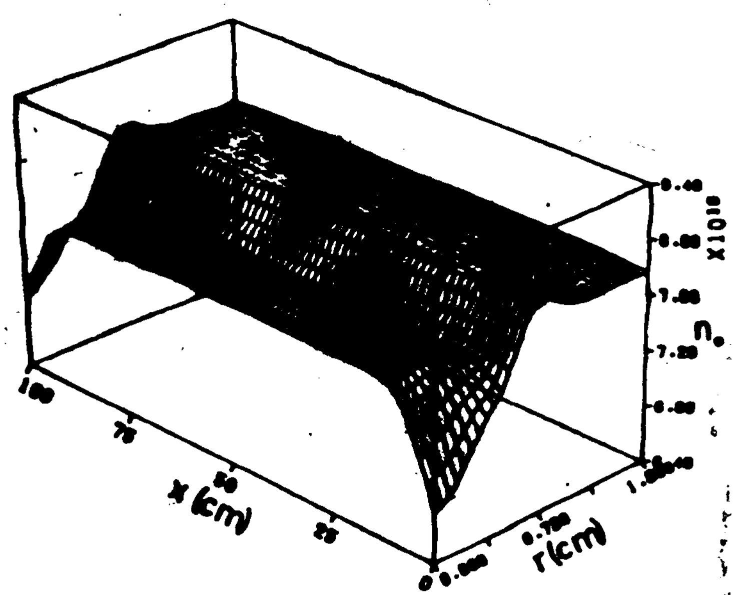

Figure 3.14 Plasma density in cm^{-3} at 0.5 μsec; obtained from the first simulation reported in Section 3.2.

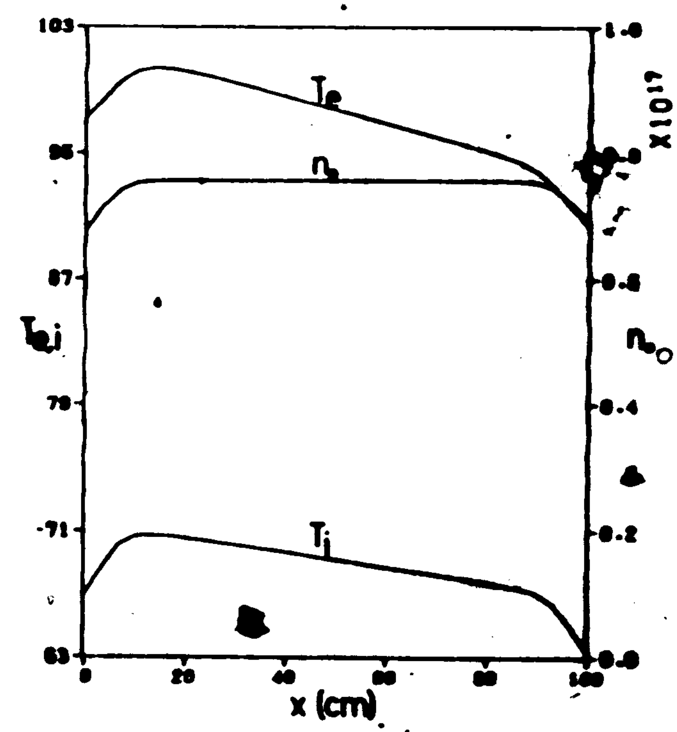

Figure 3.15 Axial values of electron temperature and ion temperature in eV and density in cm^{-3} as a function of x; obtained from the first simulation reported in Section 3.2

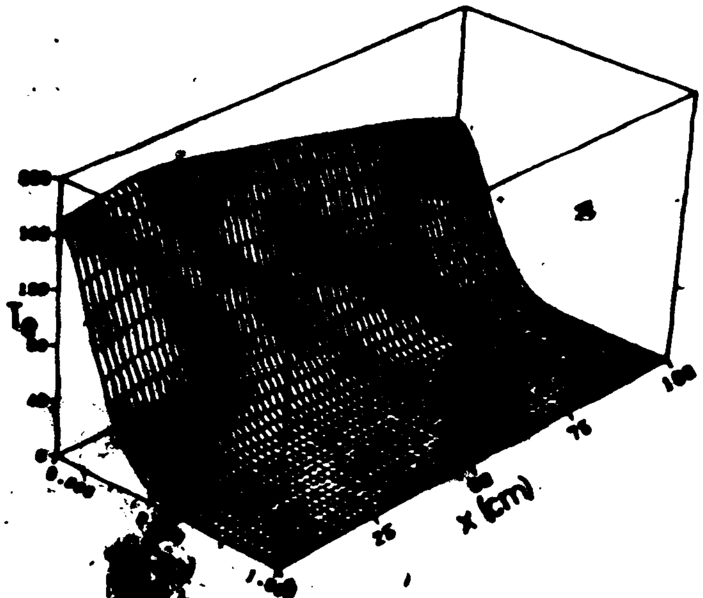

Figure 3.16 Electron temperature in eV at 1 μsec; obtained from the second simulation reported in Section 3.2.

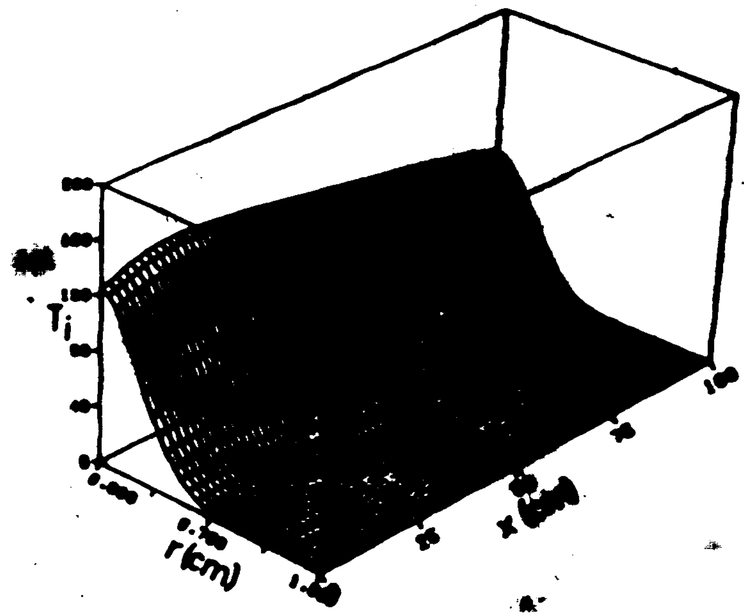

Figure 3.17 Ion temperature in eV at 1 μsec; obtained from the second simulation reported in Section 3.2.

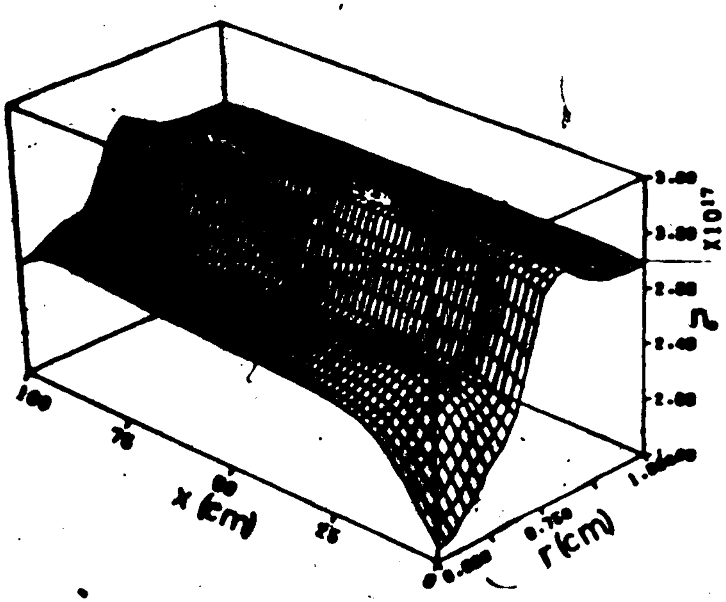

Figure 3.18 Plasma density in cm^{-3} at 1 μsec; obtained from the second simulation reported in Section 3.2.

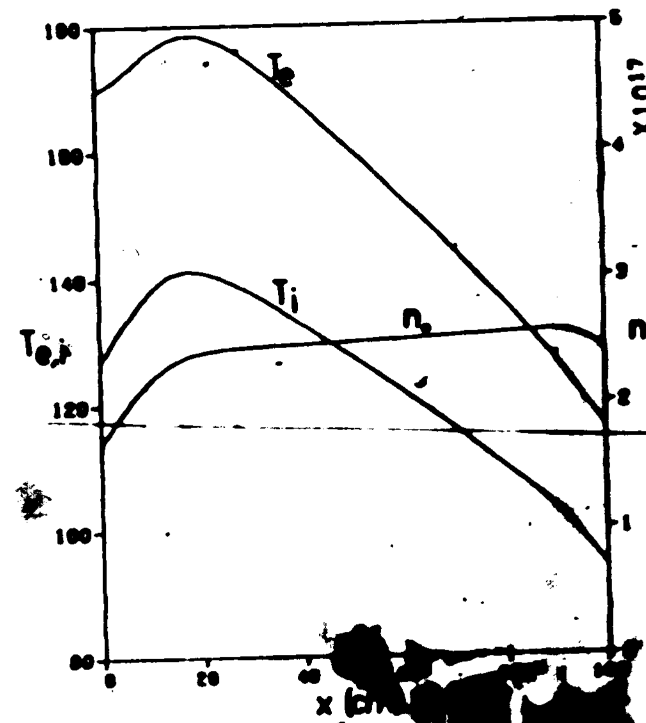

Figure 3.19 Axial values of [illegible] temperature and [illegible] temperature in eV [illegible] in cm^{-3} as a function of x; obtained from the second simulation reported in Section 3.2.

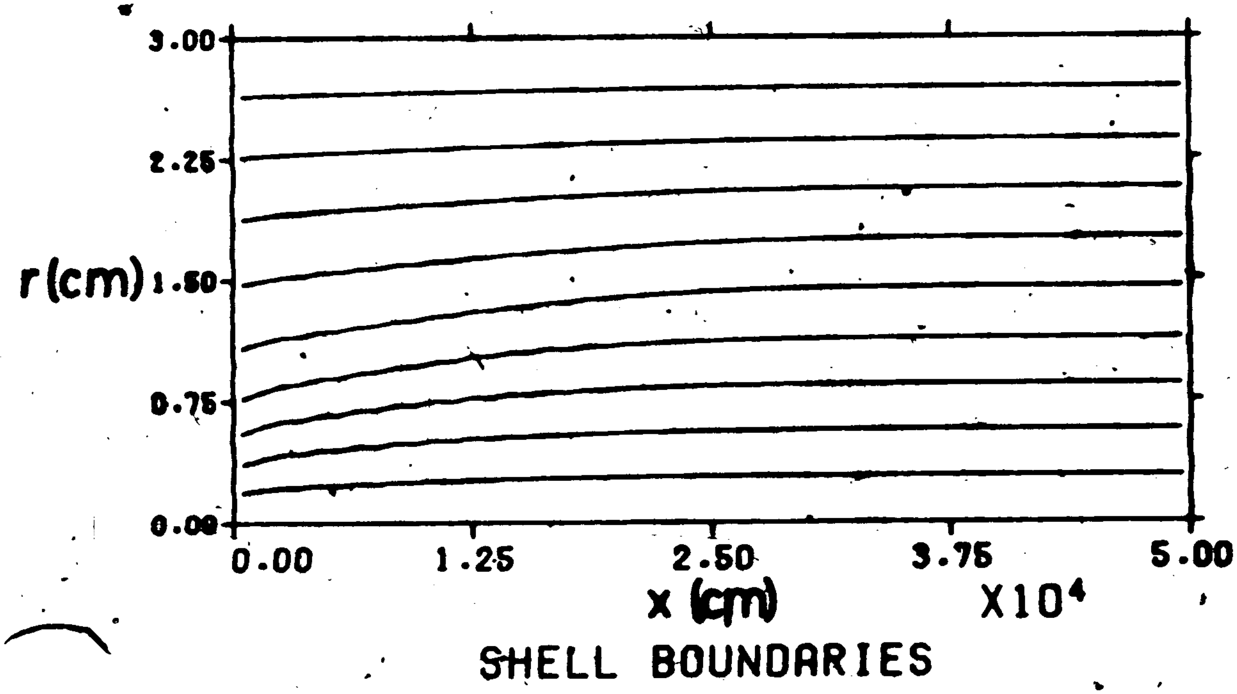

Figure 3.20 Radii of magnetic flux surfaces at 0.5 ms; from the simulation of θ-pinch reactor.

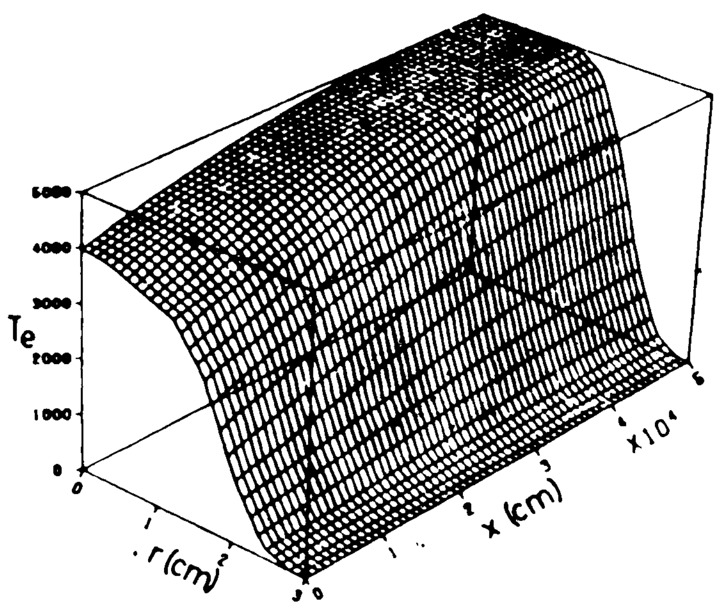

Figure 3.21 Electron temperature in eV at 0.5 ms; from the simulation of θ-pinch reactor.

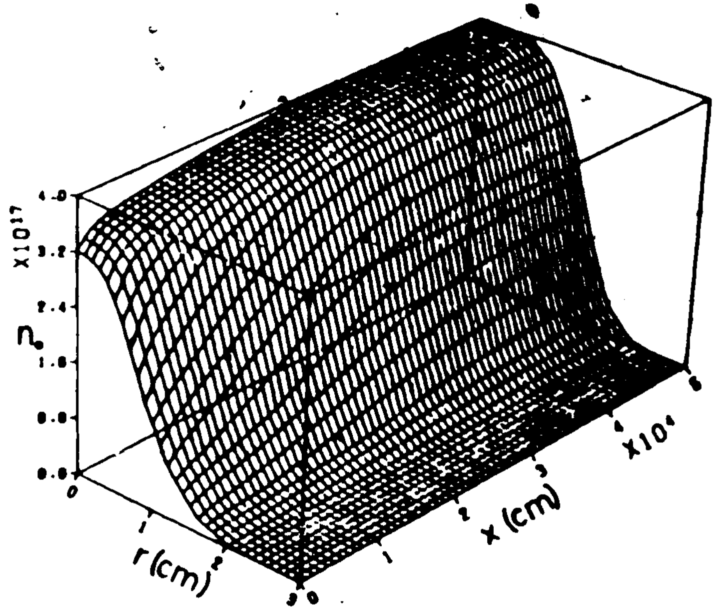

Figure 3.22 Plasma density in cm^{-3} at 0.5 ms; from the simulation of θ-pinch reactor.

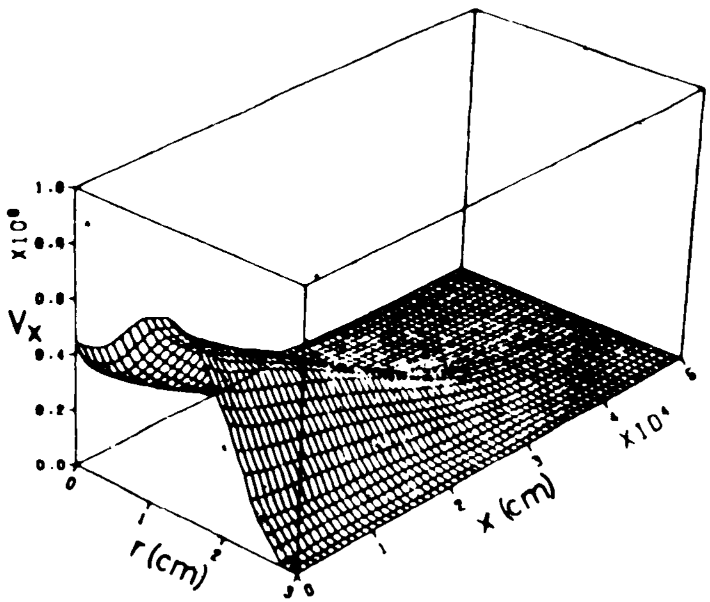

Figure 3.23 Axial plasma velocity in cm/sec at 0.5 ms; from the simulation of θ-pinch reactor.

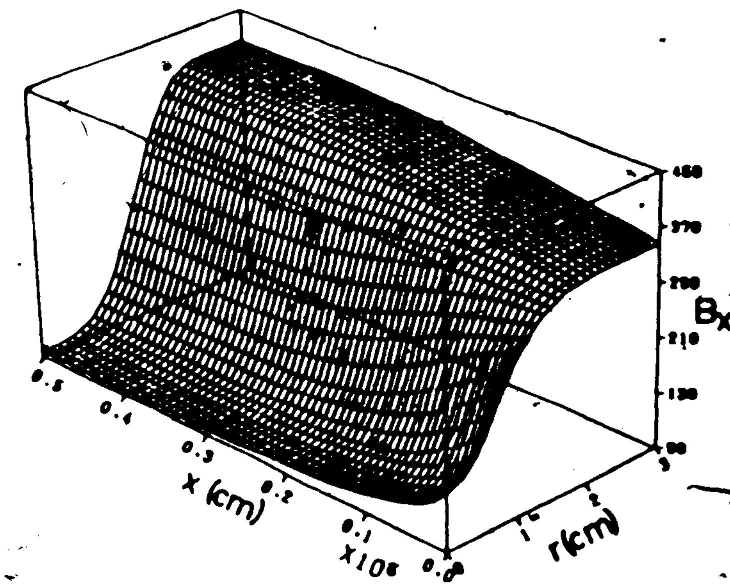

Figure 3.24 Axial magnetic field strength in KG at 0.5 ms; from the simulation of θ-pinch reactor.

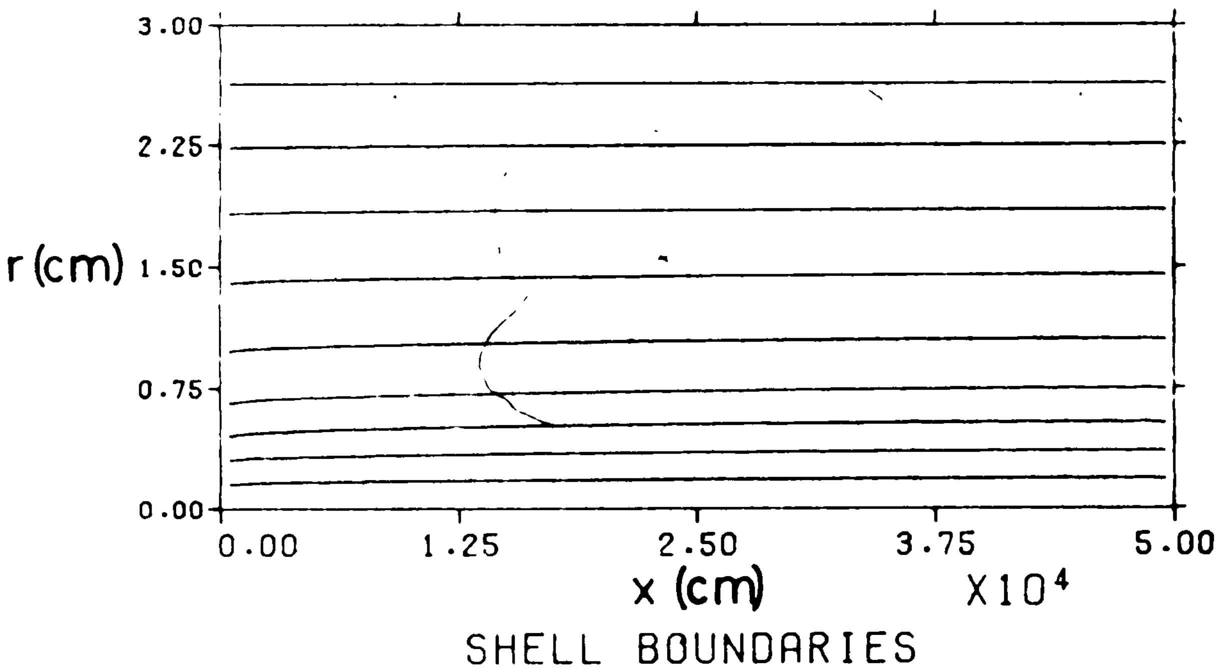

Figure 3.25 Radii of magnetic flux surfaces at 2.0 ms; from the simulation of θ-pinch reactor.

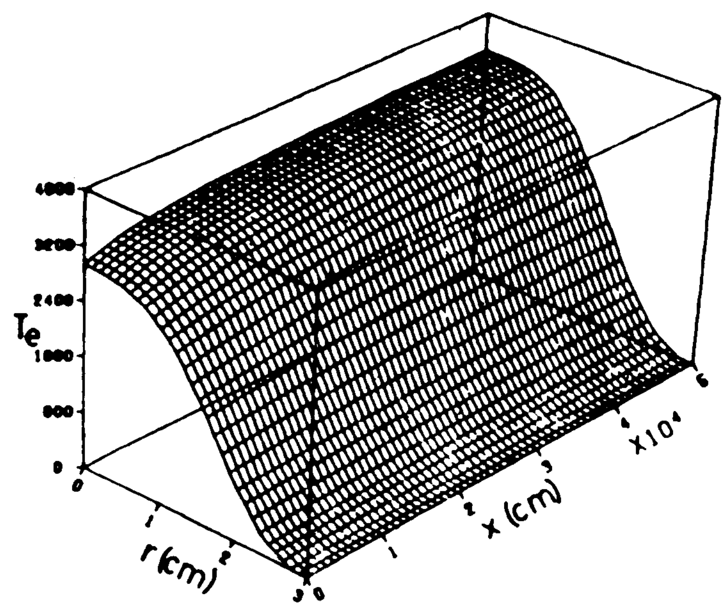

Figure 3.26 Electron temperature in eV at 2.0 ms; from the simulation of θ-pinch reactor.

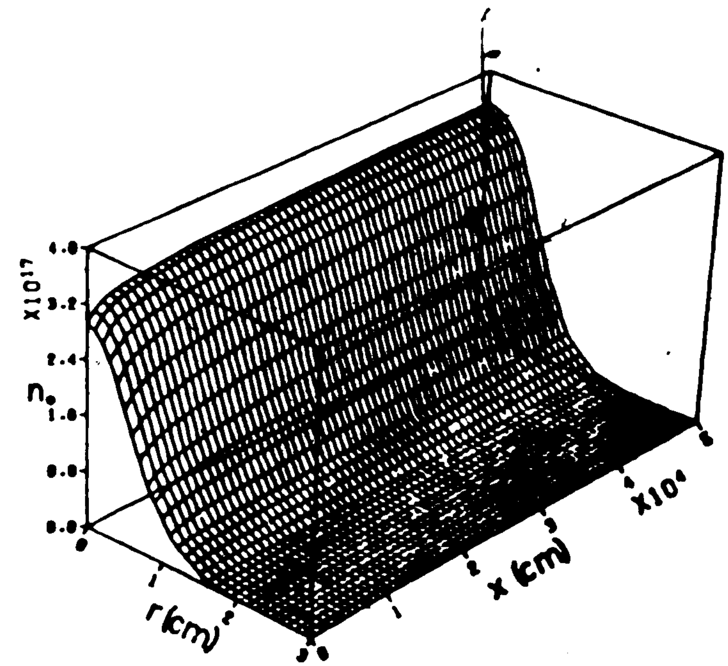

Figure 3.27 Plasma density in cm^{-3} at 2.0 ms; from the simulation of θ-pinch reactor.

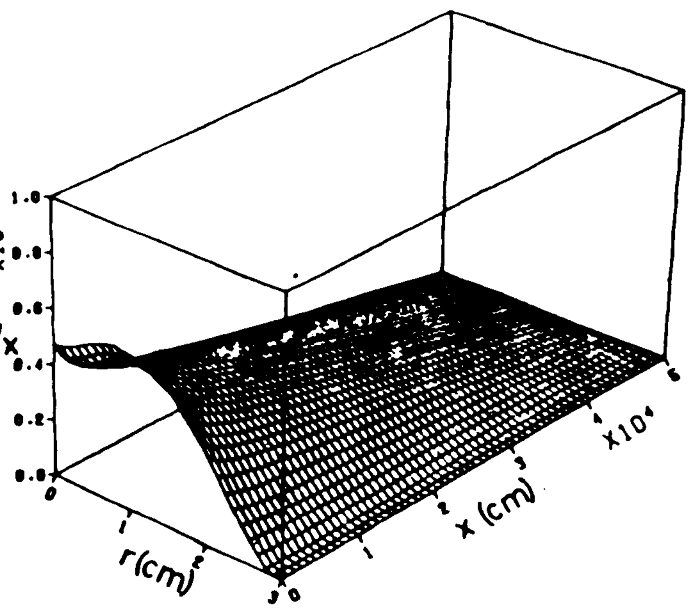

Figure 3.28 Axial plasma velocity in cm/sec at 2.0 ms; from the simulation of θ-pinch reactor.

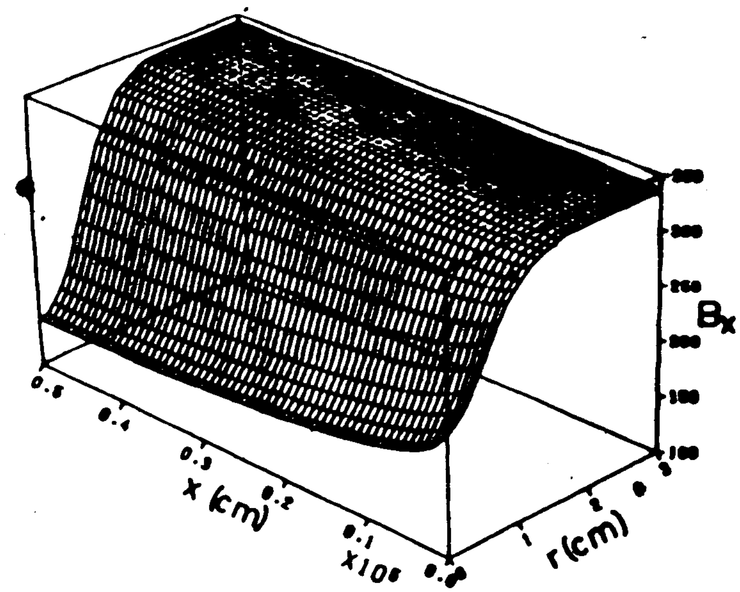

Figure 3.29 Axial magnetic field strength in KG at 2.0 ms; from the simulation of θ-pinch reactor.

CHAPTER 4

A Computer Simulation of Gas Target Experiments

Laser-induced parametric instabilities in gas target plasmas have recently been studied experimentally by A.A. Offenberger et al.[19-21] and J.J. Schuss et al.[22]. Three main advantages of using gas targets over solid targets for the study of laser-induced parametric instabilities can be summarized from apaper by A. Ng et al.[23] 1) The nonlinear interactions can be studied at controllable plasma densities which can range all the way from critical to well subcritical. 2) Nonlinearities in underdense plasmas may not be obscured by a critical layer as in solid target experiments. 3) The relatively long characteristic scale lengths in gas target experiments (as compared to solid target experiments) make spatial and temporally resolved diagnostics more accessible.

In this chapter some results from two dimensional hydrodynamic simulations of CO_2 laser heated gas target plasmas are reported. These models can be a valuable supplement to experiments since they give detailed predictions for the spatial and temporal profiles of the various hydrodynamic variables calculated. Since these quantities are very difficult to measure with good spatial and temporal resolution the simulations can be very valuable in the interpretation of experimental results. An attempt has been made to make the simulations correspond as closely as possible to the experiments of A.A. Offenberger et al.[19-21]. The experimental details of this gas target can be found in the paper by A. Ng et al.[22].

4.1 Computer Model

The computer routine used for these simulations has been described in Chapter 2. This program was developed for the purpose of modelling the magnetohydrodynamic behaviour of a plasma with an imbedded solenoidal magnetic field. For reasons described in Chapter 2 the appropriate MHD equations are solved in a moving coordinate system defined by magnetic field lines, consequently the routine will only model a plasma with an imbedded solenoidal magnetic field. However, in the gas target experiments no magnetic field exists. This problem is easily overcome by setting the solenoidal field to a sufficiently small value that it has essentially no effect on the plasma dynamics. Also, since the routine works best for (B_r/B_x) small, where B_r is the magnetic field in the radial direction, and B_x is the magnetic field in the axial direction, the electrical conductivity in the model is artificially set to a small value so that movement of the field lines is small and the factor (B_r/B_x) remains very small.

In the simulations the laser beam intensity is assumed to be of the form:

$$I(x,r,t) = \frac{P(x,t)}{2\pi\sigma(x)^2} \exp\left[- \frac{r^2}{2\sigma(x)^2} \right] \qquad (4.1)$$

where P(x,t) is the calculated total laser power. P(x,t) is calculated through a numerical solution to equation (2.10). The beam width is defined through the parameter $\sigma(x)$ which has been defined to reflect the beam focussing in free space. Refraction of the beam by the plasma has been neglected in the present model. The parameter $\sigma(x)$ has been set to conform to the optics of A.A. Offenberger *et al.* :

$$\sigma(x) = \sigma_o + \frac{1}{2f} |x - L_3| \qquad (4.2)$$

where σ_o gives the beam radius at the focal spot, f is the f number of the optics (which is set to 2 for these simulations), and L_3 specifies the position of the focal spot. The laser power input, as a function of time, is assumed to be τ_L sec long and of triangular shape. The pulse starts at zero, [illegible] reaches peak power P_o in 1/2 τ_L secs and falls off to zero again at τ_L.

In these simulations the appropriate differential equations are solved numerically in the cylindrical region defined by the following three boundaries: $r = R_o$, $x = L_o$, and $x = 0$. The initial density profile is assumed to be defined as:

$$n_o(r,x,\ t=0) = N_o\left[\ 1 + \exp\left(\frac{L_1 - x}{\lambda}\right)\right]^{-1}$$

where L_1 defines the position at which the density rises to half of N_o and λ defines the scale length over which the density rises.

For these simulations an attempt has been made to place the boundaries sufficiently far away from the region of interest that the boundary conditions do not play a significant role in the evolution of the solution. In fact R_o and L_o are sufficiently large that no plasma motion reaches these boundaries during the simulations. This is not true however for the x=0 boundary. Here we make the boundary condition that all quantities are linear ($\partial^2/\partial x^2 = 0$) with two exceptions. 1) Thermal conductivity is set to zero at x = 0. This is reasonable since there is small amount of plasma in the region x < 0 to absorb the heat.

2) $\partial P/\partial x = C$ where P is the plasma pressure ($Nk_B(T_e + T_i)$) and C is a pre-set constant. We find the solution is very insensitive to the value of C chosen.

In this chapter results from four different simulations are presented. The input parameters for the computer program are specified below. For all the simulations the plasma is assumed to be pre-ionized with an initial temperature of 1 eV. All of the simulations assume $N_o = 9\times10^{18}$ cm^{-3} and $\lambda = 0.0145$ cm. These two parameters lead to the initial density profile rising from 3% of N_o at (L_1 - 0.05) cm to 97% of n_o at (L_1 + 0.05) cm as illustrated graphically in Figures 4.1 and 4.2. The initial plasma density of 9×10^{18} cm^{-3} is just below the critical density for 10.6 μm wavelength radiation from a CO_2 laser. Other input parameters for the four simulations are:

Simulation #1	P_o = 0.1GW,	L_3 = 0.2cm,	σ_o = 0.0125cm,
	L_1 = 0.15cm,	R_o = 0.3cm,	L_o = 0.6 cm, and $\tau_L = 40\times10^{-9}$sec
Simulation #2	P_o = 1.0GW,	L_3 = 0.2cm,	σ_o = 0.0125 cm,
	L_1 = 0.15cm,	R_o = 0.5cm,	L_o = 0.7cm, and $\tau_L = 40\times10^{-9}$sec
Simulation #3	P_o = 1.0GW,	L_3 = 0.3cm,	σ_o = 0.0125cm,
	L_1 = 0.15cm,	R_o = 0.5cm,	L_o = 0.7cm, and $\tau_L = 40\times10^{-9}$sec
Simulation #4	P_o = 0.3GW,	L_3 = 0.2cm,	σ_o = 0.0035cm,
	L_1 = 0.05cm,	R_o = 0.3cm,	L_o = 0.3cm, and $\tau_L = 30\times10^{-9}$sec

In the first three simulations an attempt has been made to match the experimental conditions of A.A. Offenberger <u>et al</u>. when a stable resonator was used on the CO_2 laser. When an unstable resonator was used the angular divergence of the output beam was somewhat smaller. This leads to a smaller laser beam radius at the focal spot. In simulation #4 σ_o has been set to 35 μm to see what effect this may have on the hydrodynamics.

In all of these simulations a two-dimensional computational grid consisting of thirty radial shells and sixty axial points was used. The small value of σ_o in simulation #4 leads to very high laser intensities and therefore very high heating rates in a very small region in the vicinity of the focal spot. Because of this a finer computational mesh is needed for this simulation. This is achieved by making the region of solution smaller (smaller values of L_o and R_o) and concentrating the shell structure near the solenoid axis. That is the shell spacing is smaller for r close to zero than it is for r close to R_o. The first three simulations ran with a timestep size of close to $5x10^{-11}$ sec and required approximately 20 minutes CPU time on the Amdahl 470 V/6 computer. Simulation #4 had to be run with a much smaller timestep size (down to $2x10^{-12}$ sec as the laser beam bleached through the focal spot) and required a total of 70 minutes CPU time.

The peak laser intensity in the 4'th simulation is much higher than in the first three simulations due to the smaller value of σ_o. This can be very desirable for the experimental investigation of laser-induced parametric instabilities, however, it adds complications to the simulation of the hydrodynamic expansion since ponderomotive forces now become significant. Following the derivation of Chen[24] the ponderomotive force can be expressed as:

$$\vec{f}_p = -\frac{1}{2}\frac{\omega_{pe}^2}{\omega_1^2}\vec{\nabla}\left(\frac{I}{c_1}\right)$$

where ω_1 is the frequency of the laser beam, I is its intensity, and c_1 its group velocity. The ponderomotive force is added to the hydrodynamic model by adding the term:

$$f_r = -\frac{1}{2}\frac{\omega_{pe}^2}{\omega_1^2}\frac{\partial}{\partial r}\left(\frac{I}{c_1}\right)$$

to the right hand side of equation (2.2) and adding the term:

$$f_x = -\frac{1}{2}\frac{\omega_{pe}^2}{\omega_1^2}\frac{\partial}{\partial x}\left(\frac{I}{c_1}\right)$$

to the right hand side of equation (2.3). The appropriate modifications have been made to the code for simulation #4.

4.2 Results

The results of these simulations are summarized graphically in Figures 4.3 to 4.30. The plots are of three types. (1) At a specified time three dimensional plots of the hydrodynamic quantities (T_e, T_i, n_o, v_x and v_r) are plotted as functions of r and x. (2) At the same time the laser power P(x) and the axial laser intensity I(o,x) are plotted as functions of x. (3) The values of T_e, T_i, and n_o at the laser focal spot are plotted as functions of time.

In the three dimensional plots the density peak in the shock fronts has been limited to values of less than $2 \times 10^{19} cm^{-3}$ in order to make scaling convenient. This is achieved by setting n_o to $2 \times 10^{19} cm^{-3}$ if it exceeds $2 \times 10^{19} cm^{-3}$ in the plot routine.

The three dimensional plots of plasma density all show a "hole" in the density profile around the region of laser heating. A strong axial shock is seen to propagate just in front of the laser beam, and strong shocks are also seen to be propagating in the radial direction. The radial shocks propagate outwards with a velocity of about 4×10^6 cm/sec when the peak laser power is 0.1 GW and at about 6.5×10^6 cm/sec when peak laser power is 1.0 GW. The velocity of the axial shock is seen to be dependent on the laser beam width but a typical shock velocity of about 7.5×10^6 cm/sec is found for the case of the 0.1 GW laser. The higher powered laser beam is seen to push the axial shock at a higher velocity of between 10^7 and 2×10^7 cm/sec.

Plots of the density profiles show a hump in the region just behind the shock wave. The formation seems to be dependent on the beam width as indicated by the fact that it is found to develop later in the third simulation, where the laser is focussed further into the plasma, than it does in the second simulation. The fact that the hump is spread over about seven axial points is an indication that this feature is real and not due to the discrete numerical model.

An important feature of gas target experiments is the long characteristic scale lengths over which hydrodynamic quantities vary significantly. Density plots show that at 20 ns the density changes by 50% over axial distances of about 0.05 cm or radial distances of about 0.1 cm for the 0.1 GW laser. For the simulations with the higher powered lasers both the radial and axial scale lengths are doubled. Characteristic scale lengths for both T_e and T_i are considerably longer than for density.

The temperature plots show that electron temperature rises very rapidly to its maximum value in the region just behind the axial shock. This can be explained by the high laser absorption rate in the high density region at the back end of the shock. Further back the laser heating rate is reduced by the lower densities and higher electron temperatures, and must compete with the cooling effect of adiabatic expansion. The ion temperature profile is similar, although the ions remain considerably colder than the electrons. The ions are heated rapidly in the high density region just behind the axial shock due to electron-ion collisions. The subsequent adiabatic expansion causes the ions to cool somewhat in the region behind the shock. Typical electron temperatures of 40 to 50 eV and ion temperatures of about 30 eV are achieved in the first simulation with the 0.1 GW laser. The second simulation shows that increasing the laser power by a factor of 10 leads to an increase in typical electron and ion temperatures by a factor of about two. Increasing the laser power by a factor of ten leads to a large change in the volume of the plasma heated rather than making a large change in the plasma temperature. The third simulation, where the laser is focussed further into the plasma, shows significantly higher electron temperatures can be achieved in the region of the focal volume if the laser beam is focussed sufficiently far into the plasma to cause an axial shock to be driven through the focal volume.

The plots of laser power versus x show that the beam power is fairly constant up to the shock wave. This shows that almost all of the laser energy is being absorbed in the high density region at the back of the axial shock.

A knowledge of plasma fluid velocities can be important in the interpretation of experiments since the fluid velocity can lead to a doppler shift in the frequency of plasma light emission. The axial velocity plots show typical axial velocities in the focal volume of about 10^7 cm/sec.

The plots of T_e, T_i, and n_o as a function of time provide a summary of how the values of these variables in the focal volume change over the duration of the experiment. It is seen that for the simulation with the 0.1 GW laser the electron temperature in the focal volume rapidly rises to 55 eV, and then slowly cools to 40 eV at 30 ns. The ion temperature, however, rises rapidly to 30 eV and maintains this value throughout the time of the simulation. The ion heating, due to electron-ion collisions, is presumably balanced by the ion cooling, due to adiabatic expansion. Increasing the laser power by a factor of ten, as in the second simulation, leads to electrons in the focal volume heating rapidly to 95 eV and then cooling to 75 eV at 30 ns. The ions rapidly heat to about 35 eV, and then continue to rise in temperature until they reach 50 eV at 30 ns. It is interesting to note that the fluid densities at the focal volume for these two simulations are almost identical over the period of 12 ns to 30 ns. These plots also show the importance of the laser focus position. In the third simulation, where the beam was focussed further into the plasma, the electron temperature rapidly rose to 155 eV as the shock wave propagated through the plasma. The plots indicate that at about 7 ns a hot plasma of critical density existed in the focal volume. Since this critical layer is part of the shock wave, characteristic scale lengths are very short at this time. After the shock passes through the focal volume, the electron temperature cooled much more rapidly than was found in the other simulations. By 30 ns the electron temperature is only 75 eV which is the same as the electron temperature of the second simulation at 30 ns. The fluid density also fell rapidly and became essentially the same as that of the second simulation by 22 ns. The ion temperature in the focal volume of the third simulation rose rapidly to about 50 eV and remained constant for the duration of the simulation. Another important feature seen in these plots is that the density curve goes through the quarter critical value fairly quickly before becoming more level at a density of about 10^{17} cm^{-3}. The time at which the density curve goes through the quarter critical value depends on both the laser power and the laser focus position.

The three dimensional plots for the fourth simulation are shown at a time of 12 ns. This is close to the time of peak laser power and is just after the laser beam has bleached its way through the focal volume. A peak electron temperature in excess of 200 eV is seen at this time while the peak ion temperature is close to 60 eV. The three dimensional plot of plasma density shows no axial shock, however other plots not included in this thesis show that an axial shock exists at both earlier and later times. The reason for this effect is that as the laser beam bleaching wave propagates through the focal volume, the bleaching velocity exceeds the acoustic velocity in the hot plasma behind the bleaching front. This high velocity bleaching wave temporarily destroys the shock wave. The plots of v_r and v_z show much larger fluid velocities in the region of the bleaching front than that of previous simulations. This is a reflection of the large pressure gradients that have been set up.

In Figure 4.30, n_o, T_e, and T_i at the position of the laser focal spot are plotted as a function of time. This plot shows that the axial shock reaches the focal spot before it is destroyed as indicated by Figure 4.27. As the back end of the shock wave passes the focal point of the laser beam, the electron temperature rises very rapidly to reach a peak of 265 eV. Just after the shock wave passes through the focal spot the plasma expands rapidly and the electron temperature falls to about 100 eV in a period of about 1 ns, and then continues to drop but at a much slower rate. This rapid cooling can be attributed to the rapid adiabatic expansion as well as thermal conduction due to large temperature gradients.

4.3 Conclusions

The computer simulations of laser heated gas target plasmas predict that the hydrodynamic quantities vary over relatively long characteristic scale lengths. Both axial and radial shocks propagate away from the region of laser heating. Most of the laser energy is found to be deposited in a small volume of plasma just behind the axial shock, leading to both electrons and ions reaching their peak temperature at a short distance behind the shock. The plasma temperature and density in the laser focal volume is found to be a relatively weak function of input laser power.

It is found that the position of the laser focal spot relative to the initial density profile can be important to the resulting hydrodynamics. If the laser is focussed sufficiently far into the initial density profile that an axial shock wave is driven through the focal volume, the early time electron temperatures and densities are found to be much higher than temperatures and densities at corresponding times for the case where the laser is focussed at the front edge of the density profile. These differences could lead to significantly different observations of laser induced parametric instabilities. The hydrodynamic behaviour of the plasma at later times (> 25 ns), however, is found to be relatively insensitive to the position of the laser focal spot.

The reduction of the beam radius at the focal spot can have dramatic effects on the resulting hydrodynamics. While the electron temperature at the focal spot rose to a much higher value for the case of σ_o = 35 μm as compared to when σ_o = 125 μm, it also decreased much more rapidly. The plasma density also drops more quickly for smaller

of σ

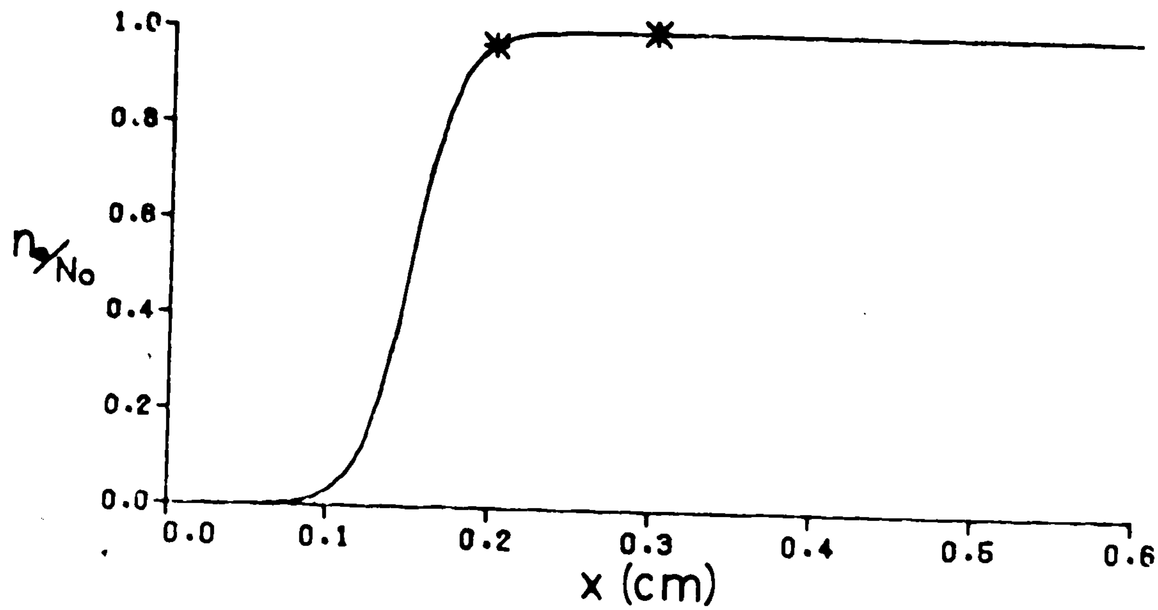

Figure 4.1 Initial axial density profile used in gas target simulations #1, #2, and #3. The left * denotes the position of laser focus for simulations #1 and #2. The right * denotes the position of laser focus for simulation #3.

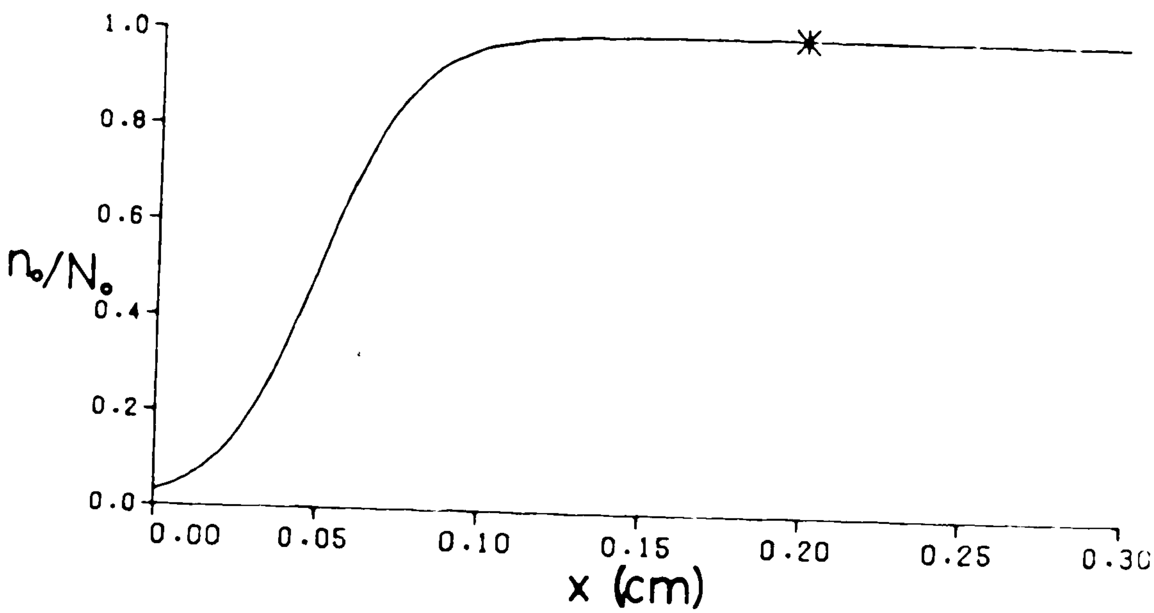

Figure 4.2 Initial axial density profile used in gas target simulation #4. The * denotes the position of laser focus.

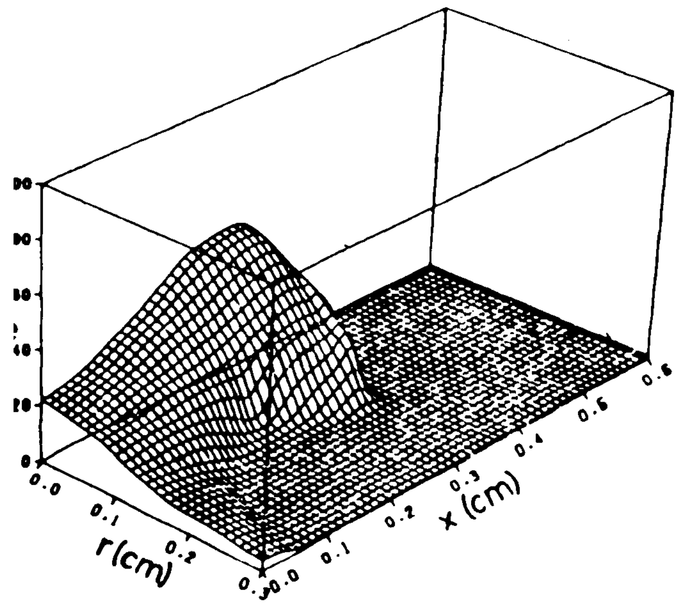

Figure 4.3 Electron temperature in eV at 20 ns; from simulation #1.

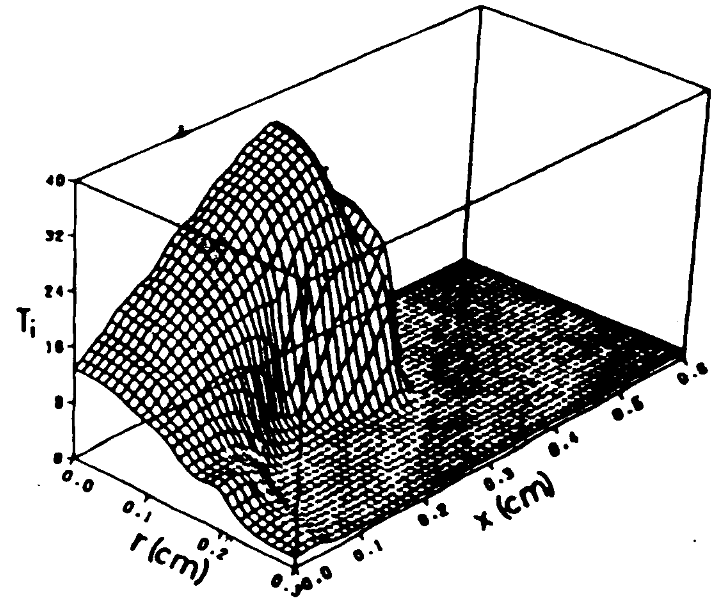

Figure 4.4 Ion temperature in eV at 20 ns; from simulation #1.

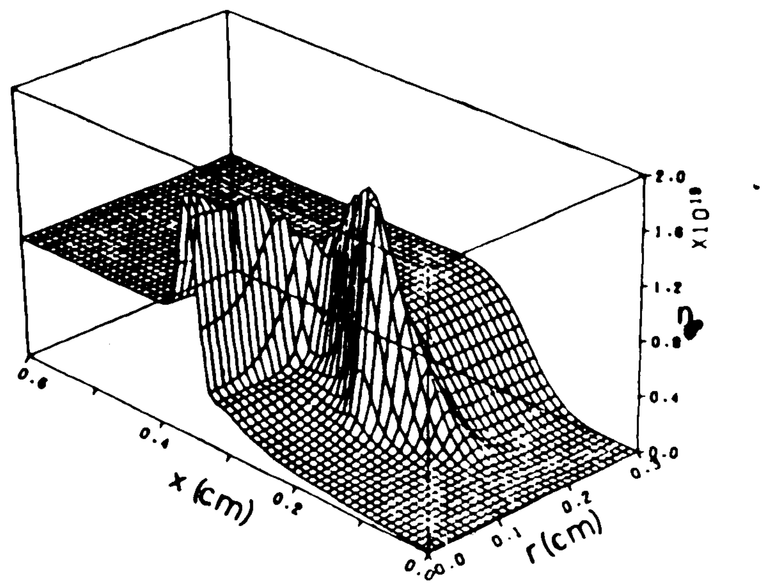

Figure 4.5 Plasma density in cm^{-3} at 20 ns; from simulation #1.

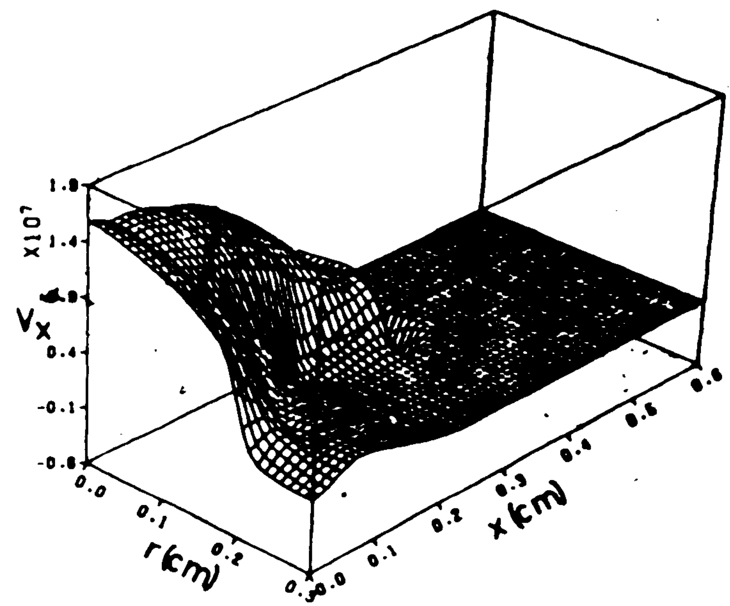

Figure 4.6 Axial plasma velocity in cm/sec at 20 ns; from simulation #1.

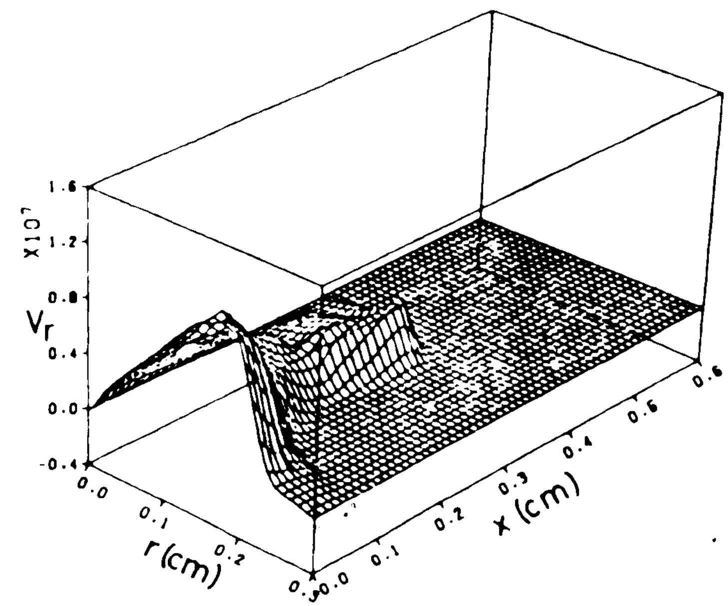

Figure 4.7 Radial plasma velocity in cm/sec at 20 ns; from simulation #1.

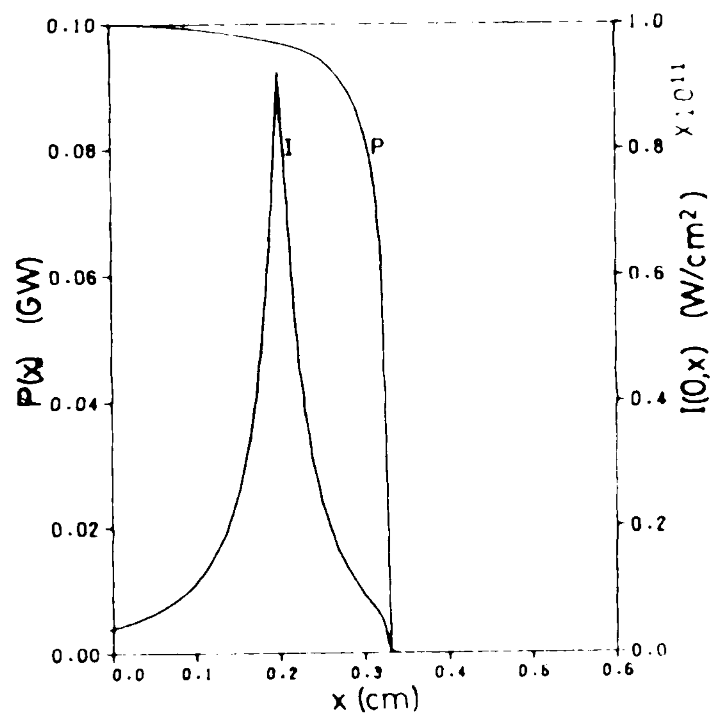

Figure 4.8 Laser beam power and axial laser beam intensity at 20 ns; from simulation #1.

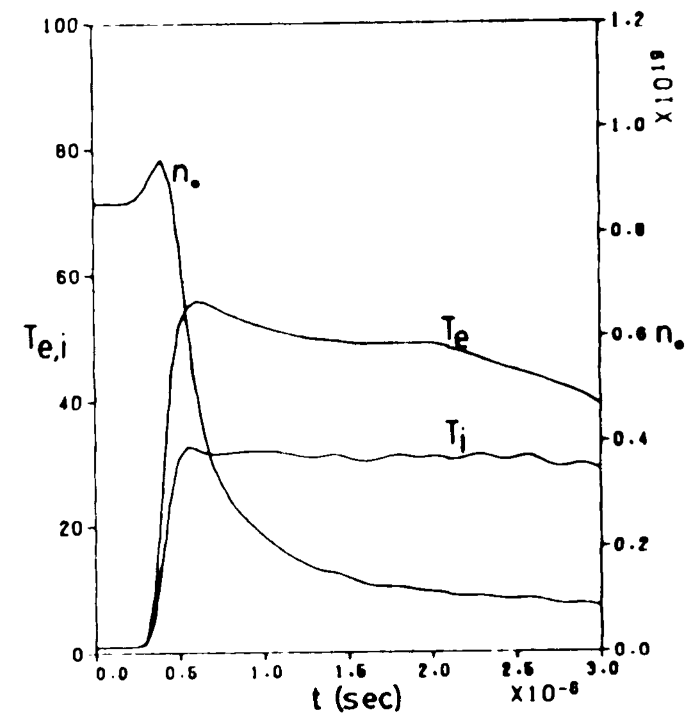

Figure 4.9 Electron and ion temperatures in eV and plasma density in cm^{-3} at the laser focal spot as a function of time; from simulation #1.

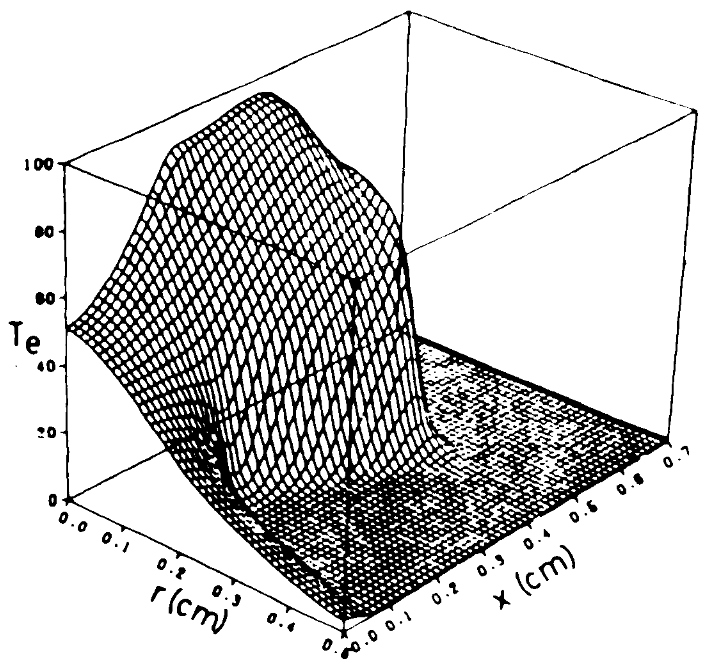

Figure 4.10 Electron temperature in eV at 20 ns; from simulation #2.

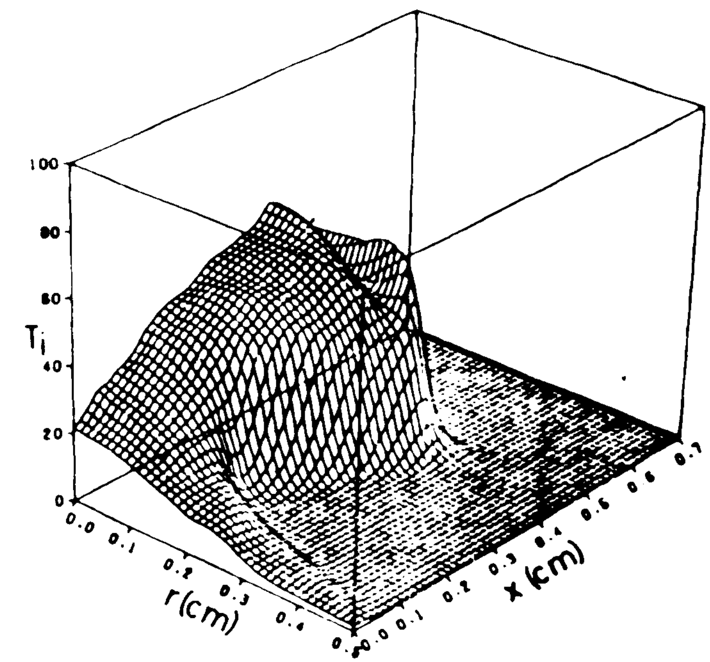

Figure 4.11 Ion temperature in eV at 20 ns; from simulation #2.

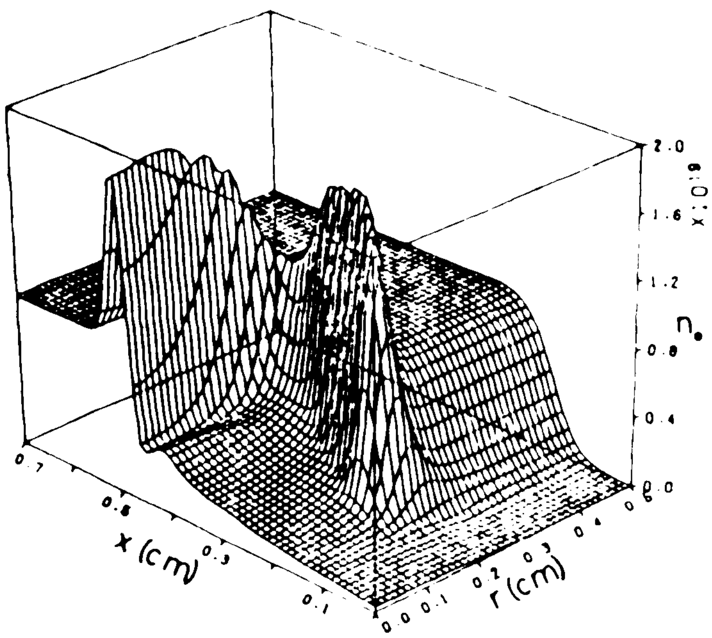

Figure 4.12 Plasma density in cm^{-3} at 20 ns; from simulation #2.

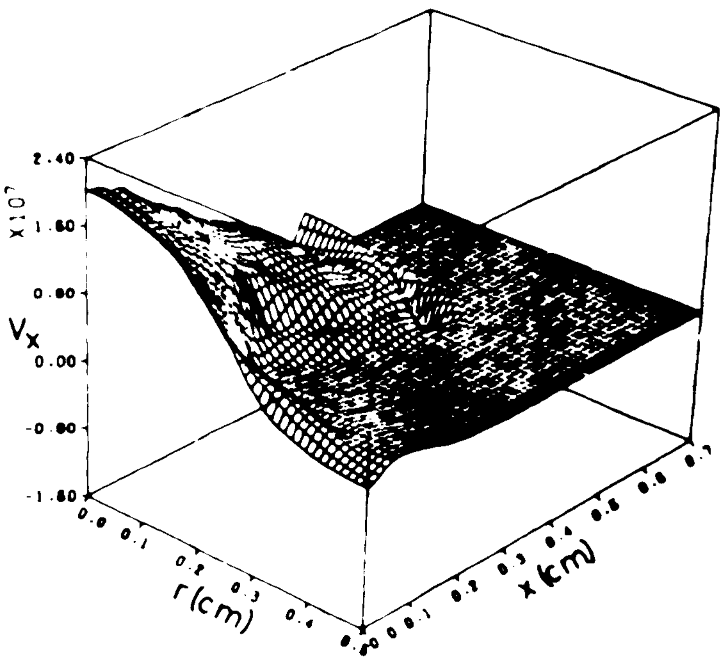

Figure 4.13 **Axial plasma velocity in cm/sec at 20 ns; from simulation #2.**

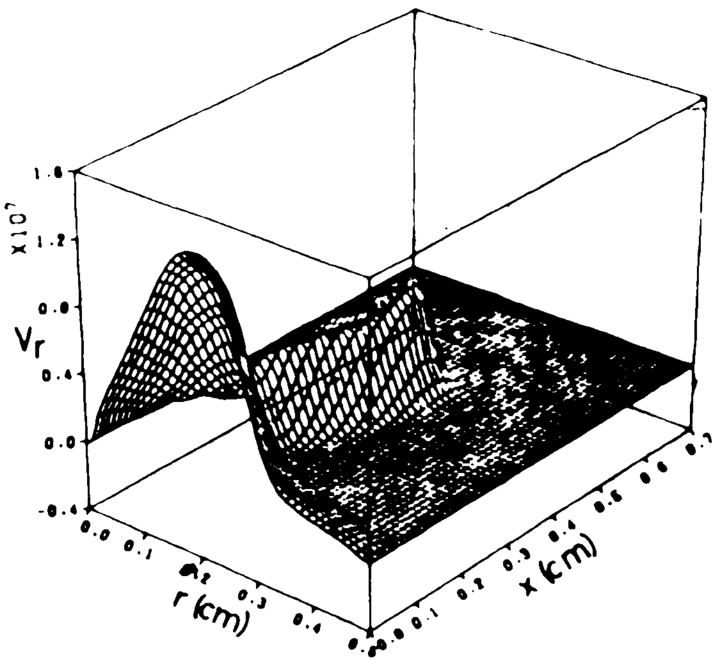

Figure 4.14 **Radial plasma velocity in cm/sec at 20 ns; from simulation #2.**

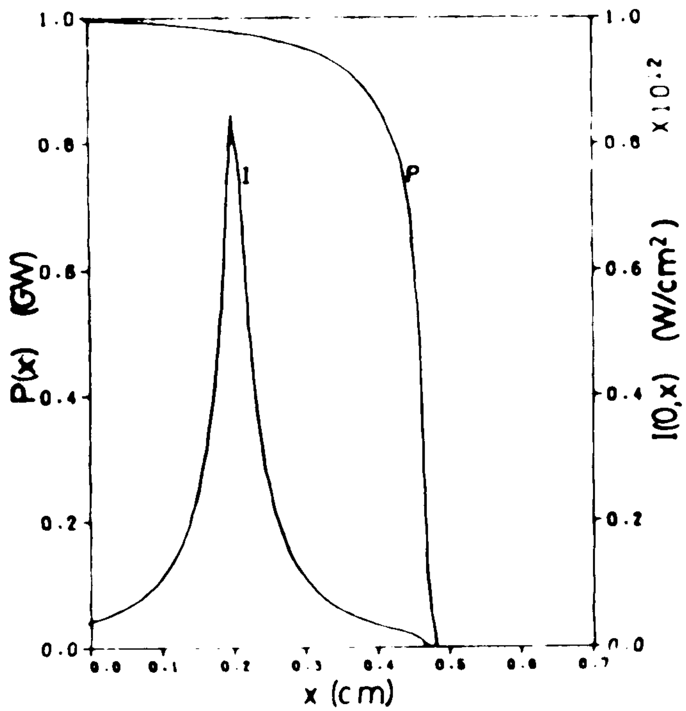

Figure 4.15 Laser beam power and axial laser beam intensity at 20 ns; from simulation #2.

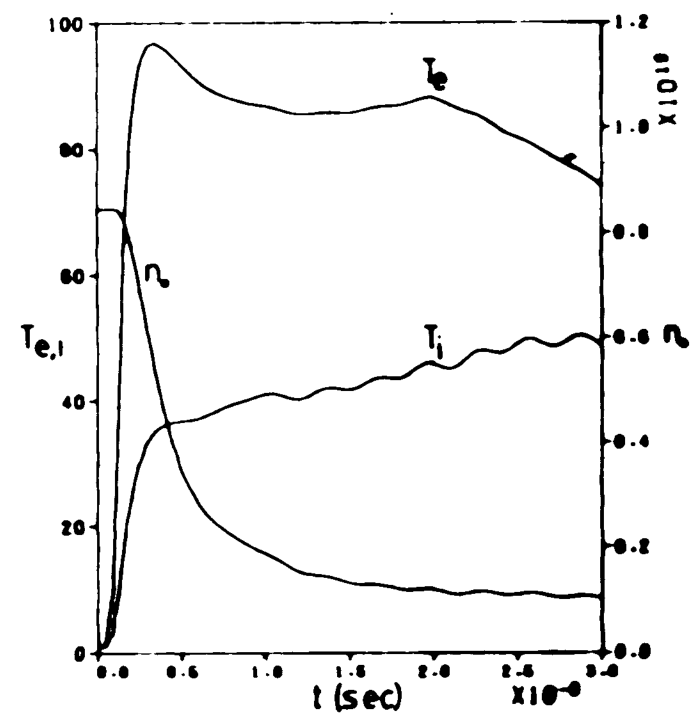

Figure 4.16 Electron and ion temperatures in eV and plasma density in cm^{-3} at the laser focal spot as a function of time; from simulation #2.

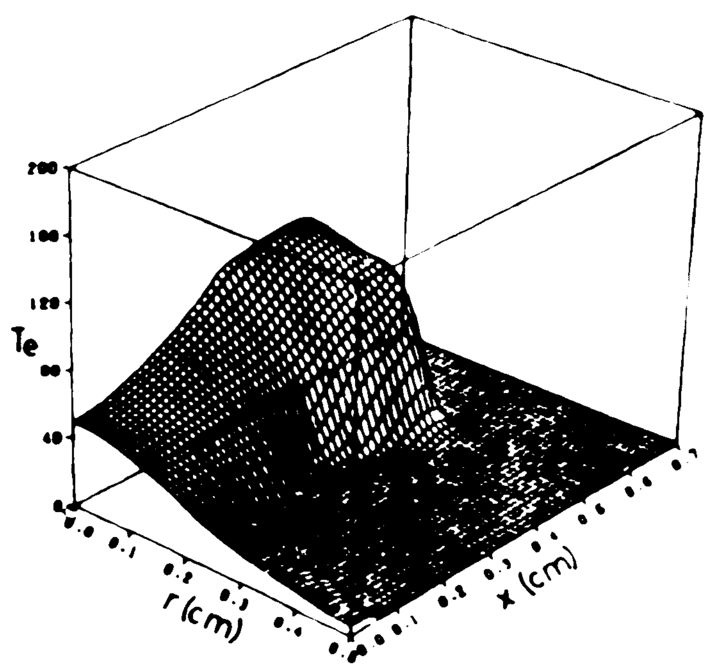

Figure 4.17 Electron temperature in eV at 20 ns; from simulation #3.

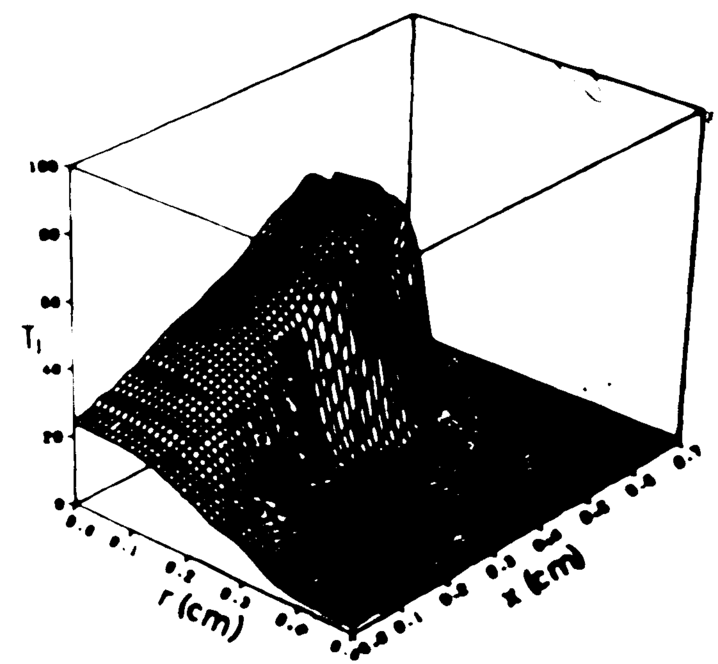

Figure 4.18 Ion temperature in eV at 20 ns; from simulation #3.

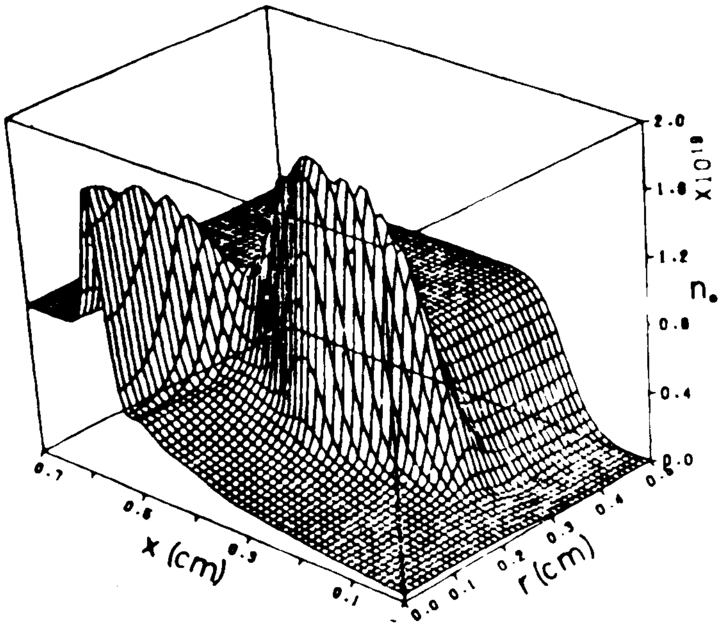

Figure 4.19 Plasma density in cm^{-3} at 20 ns; from simulation #3.

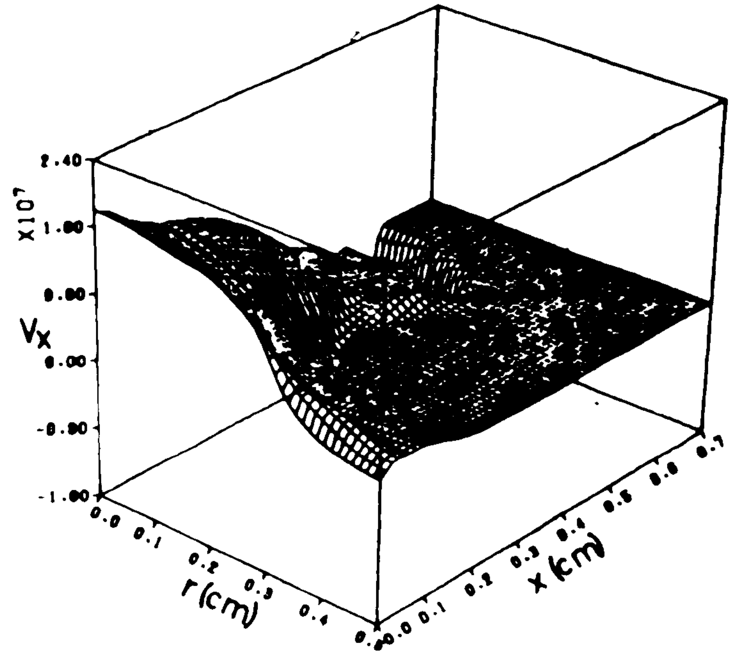

Figure 4.20 Axial plasma velocity in cm/sec at 20 ns; from simulation #3.

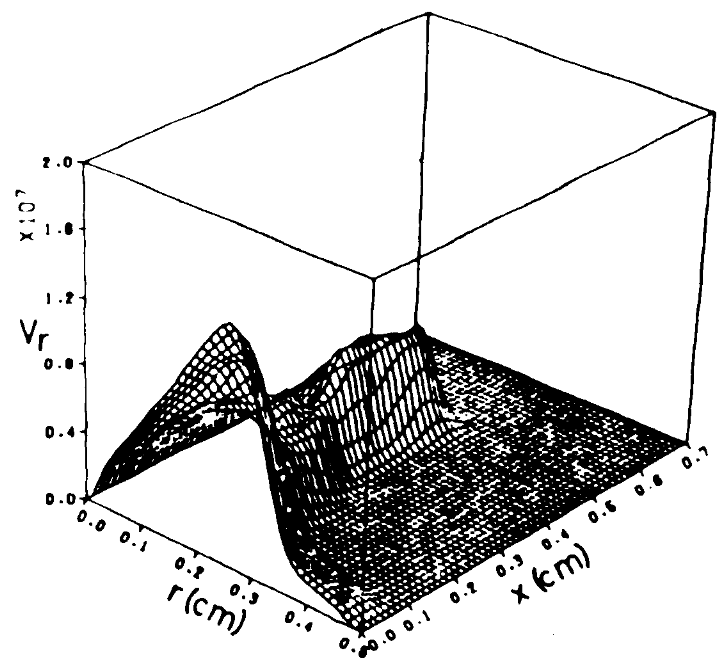

Figure 4.21 Radial plasma velocity in cm/sec at 20 ns; from simulation #3.

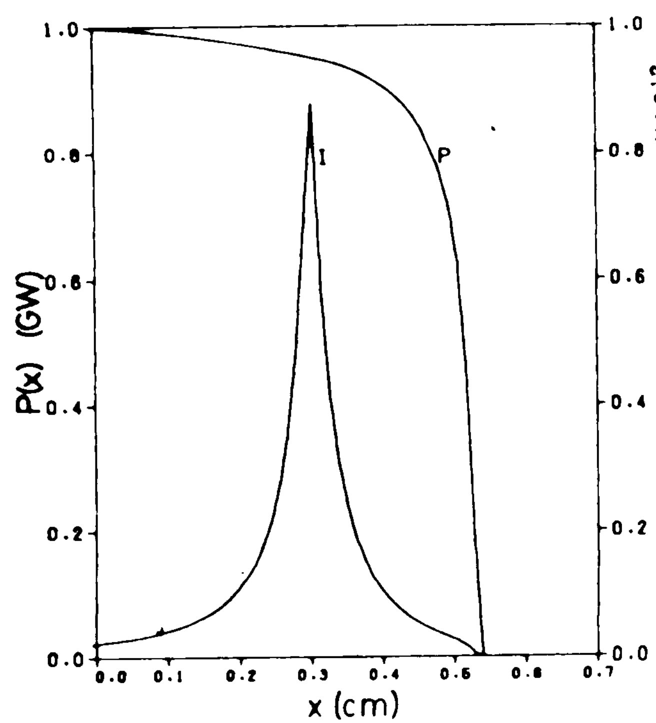

Figure 4.22 Laser beam power and axial laser beam intensity at 20 ns; from simulation #3.

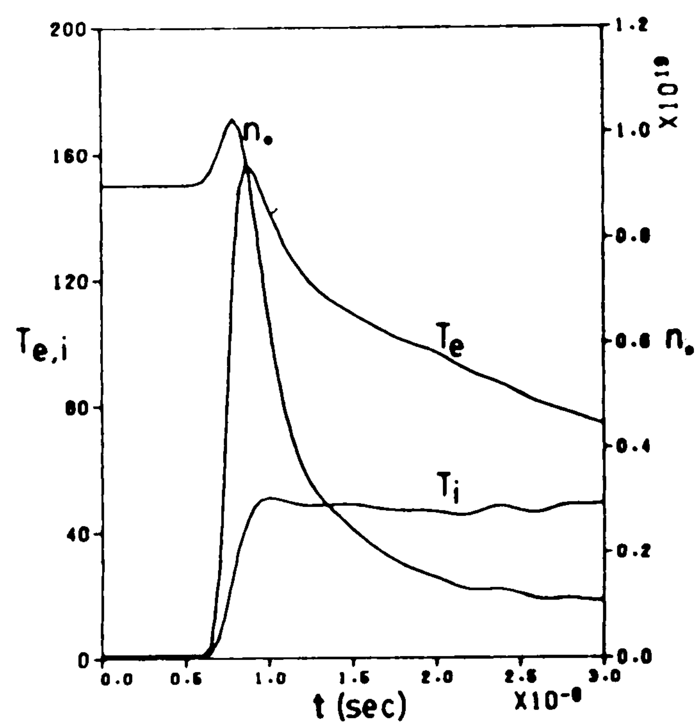

Figure 4.23 Electron and ion temperatures in eV and plasma density in cm^{-3} at the laser focal spot as a function of time; from simulation #3.

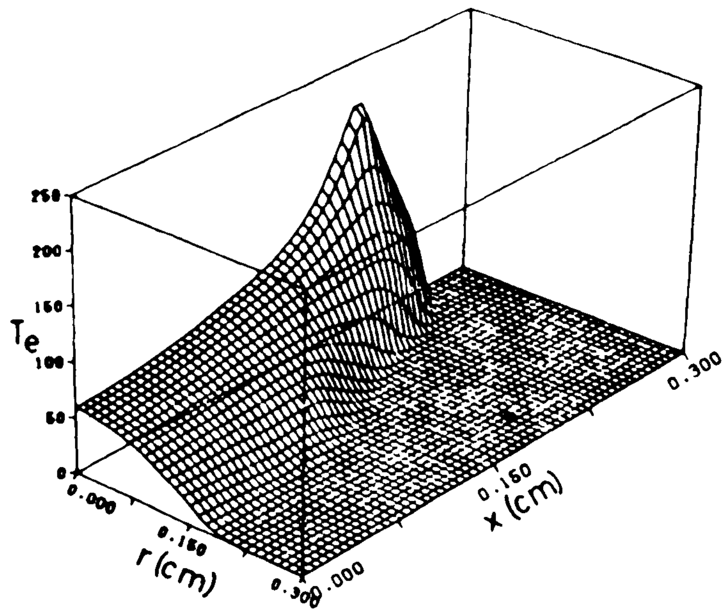

Figure 4.24 Electron temperature in eV at 12 ns; from simulation #4.

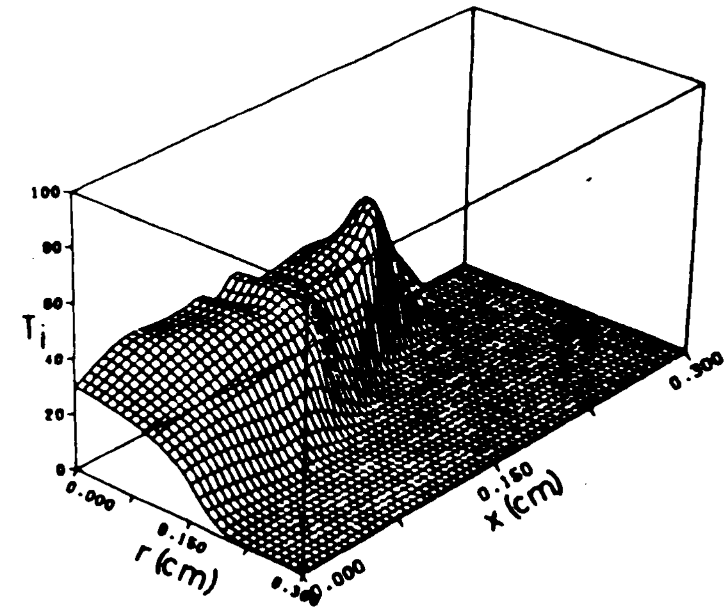

Figure 4.25 Ion temperature in eV at 12 ns; from simulation #4.

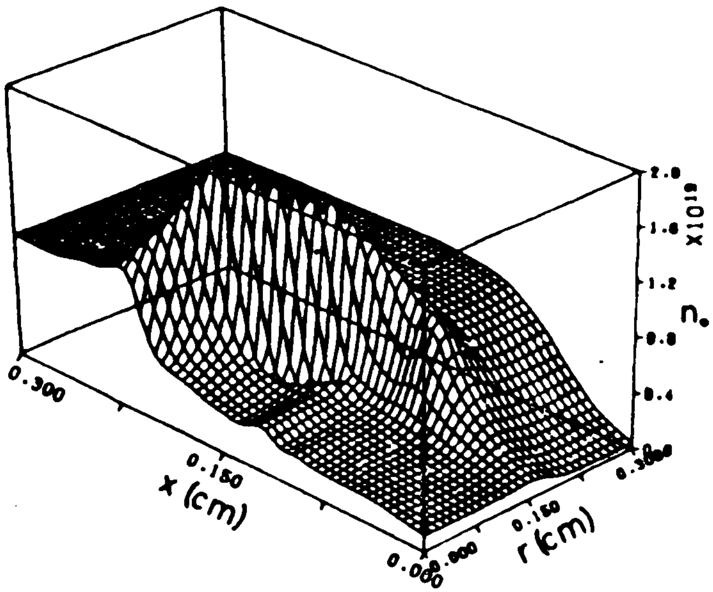

Figure 4.26 Plasma density in cm^{-3} at 12 ns; from simulation #4.

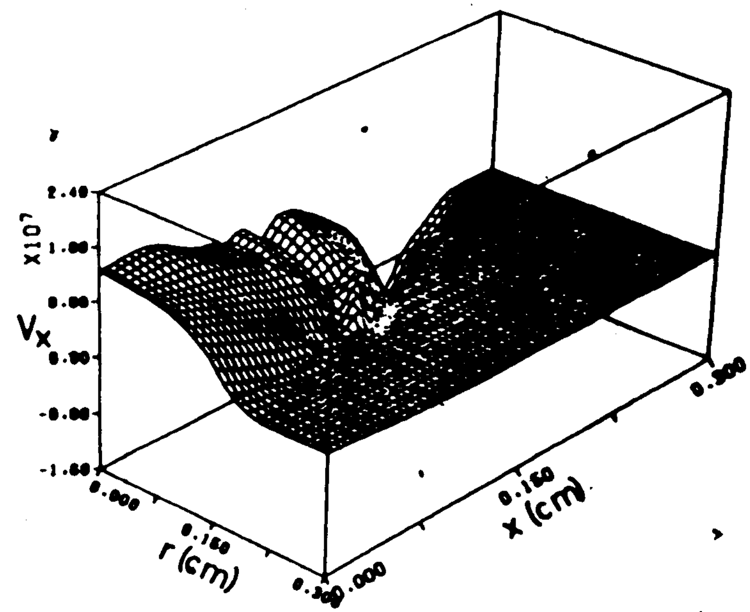

Figure 4.27 Axial plasma velocity in cm/sec at 12 ns; from simulation #4.

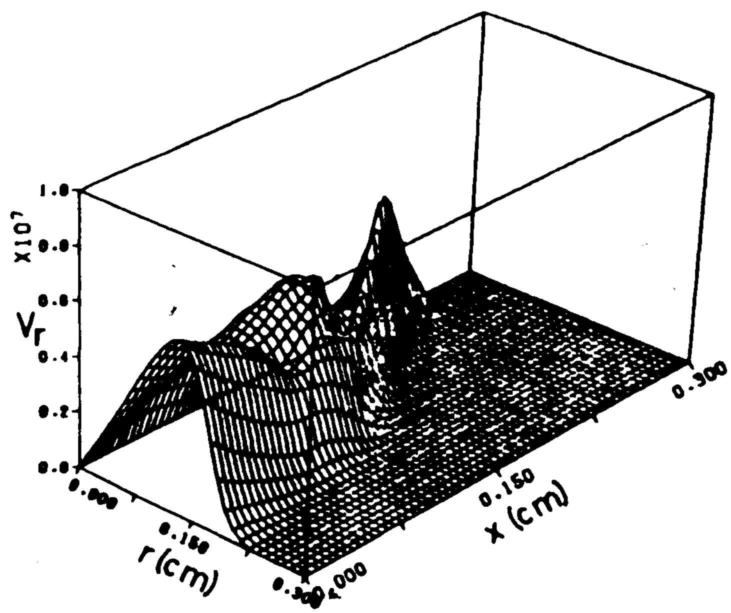

Figure 4.28 Radial plasma velocity in cm/sec at 12 ns; from simulation #4.

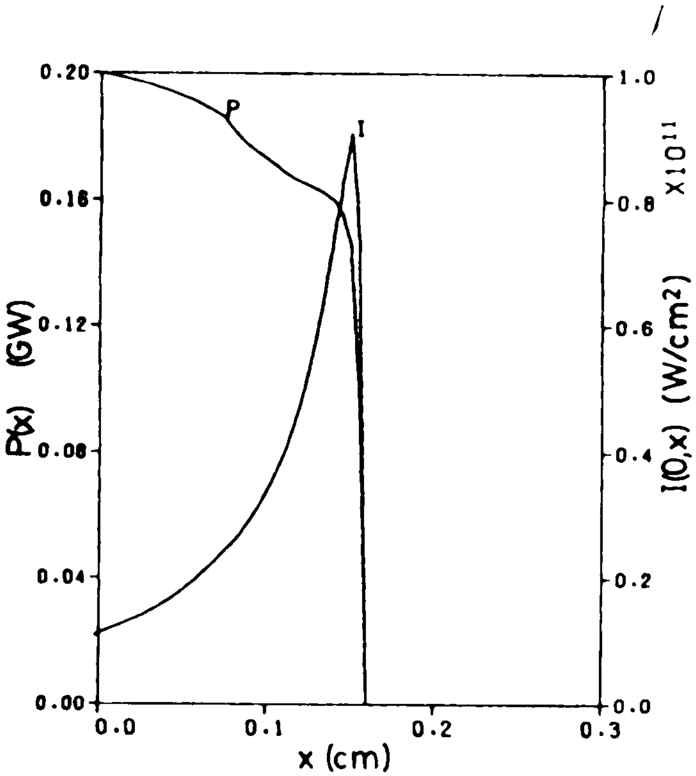

Figure 4.29 Laser beam power and axial laser beam intensity at 10 ns; from simulation #4.

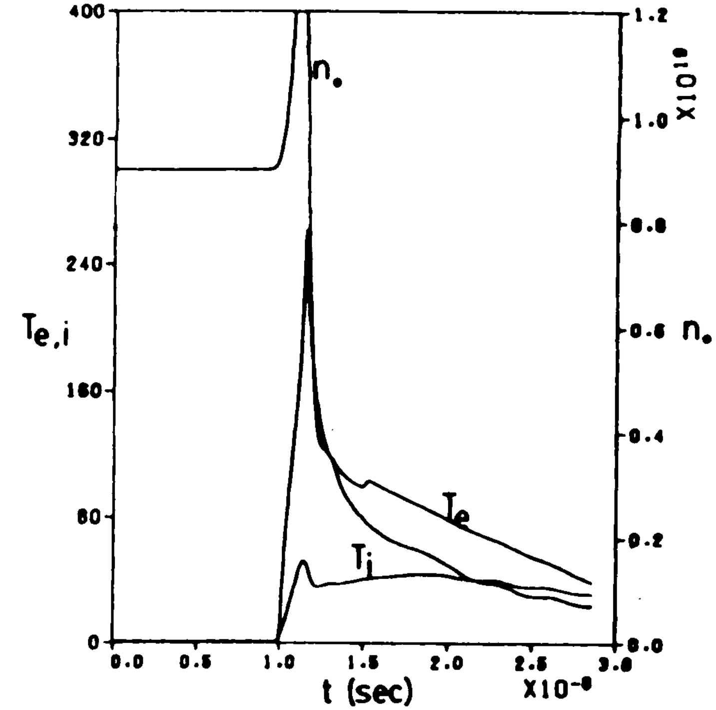

Figure 4.30 Electron and ion temperatures in eV and plasma density in cm^{-3} at the laser focal spot as a function of time; from simulation #4.

CHAPTER 5

A Theory of Beat Frequency Mixing

In this chapter a theory for the beat frequency mixing of antiparallel electromagnetic beams in a plasma is presented. This theory assumes that the plasma is uniform and has an isotropic Maxwellian distribution function. The magnetic field, if present, is assumed to be parallel to the directions of propagation of the interacting beams.

Theoretical studies on the beat frequency mixing of antiparallel electromagnetic beams, where the difference frequency of the interacting beams is close to the electron plasma frequency, have been made by Schmidt[25], Fuchs et al.[26], Kaufman and Cohen[27], Cohen[28], Cohen et al.[29], and Capjack and James[30]. Much of the work in references 25-29 is devoted to the theory of beat frequency mixing in an inhomogeneous plasma. Capjack and James present a theory for beat frequency mixing in a homogeneous plasma. It was found, however, that the results of Capjack and James differed by a factor of about $(\omega_{pe}/\omega_3)^4$ from the calculation made by Cohen et al.[29]. The source of this discrepancy is explained in this chapter.

Results from Chapter 3 of this thesis indicate that it may be possible to create a plasma which is radially confined by a solenoidal magnetic field with very long density and temperature scale lengths in the axial direction. For these con[illegible] the theory for beat frequency mixing in a homogeneous plasma is va[illegible] Capjack and James[10] have suggested that the beat frequency [illegible]parallel laser beams in a plasma radially confined by a sole[illegible] magnetic field could be used to create a useful broad-band infrared amplifier. A good theory for the beat-frequency mixing process in homogeneous plasmas is a valuable tool for the theoretical study of such a device.

In this chapter beat frequency mixing rates for a homogeneous plasma are derived. The effects of ion mobility and electron-ion collisions have been included in this derivation. Inclusion of ion mobility leads to the calculation of high mixing rates for the case when the beat frequency is close to the ion-acoustic frequency. This approach leads to an expression that is valid for difference frequencies ranging from zero to much greater than the electron-plasma frequency. Inclusion of the effects of electron-ion collisions makes the results equally valid for cases where the dominant mechanism for the damping of the driven electrostatic mode is collisional, collisionless, or the intermediate case where both effects are important.

In Section 5.1 differential equations which describe the coupling of antiparallel electromagnetic beams in the presence of electron density fluctuations are derived. This derivation follows Cohen[28] and has been included here for completeness. In Section 5.2 an integral equation which defines the electron density fluctuation caused by the beating of antiparallel electromagnetic beams is derived. The two differential equations from Section 5.1 and the integral equation from Section 5.2 form a set of three coupled equations which describe the beat frequency mixing process. In Section 5.3 these three equations are simplified through an approximation based on the assumption that the amplitudes of the fields in the electromagnetic beams vary slowly on the time and length scales defined by the beat frequency period and wavelength. Beat frequency heating rates are derived for this particular case. In Section 5.4 the equations of Section 5.1 and Section 5.2 are cast in an approximate form which allows for easy numerical solution. These new equations are useful since they can be used to describe the beat frequency mixing of a beam with a short pulse while properly accounting for transient effects in the electrostatic mode. In Section 5.5 the dependence of the beat frequency mixing rates on plasma parameters are displayed graphically and discussed.

In the derivation that follows it is assumed that the interacting beams are plane polarized, however it can be shown that all the results are equally valid for the case of circular polarization.

5.1 Beam Coupling in the Presence of Electron Density Fluctuations

In this section an expression for the coupling of two anti-parallel, linearly polarized, electromagnetic beams in the presence of electron density fluctuations is derived. Effects of collisional damping (inverse Bremsstrahlung absorption) are included. The frequency of the electromagnetic beams, ω_i $(i=1,2)$ is assumed to be greater than the electron plasma frequency, ω_{pe}.

The vector potential in the transverse gauge due to the two interacting beams can be written as:

$$\partial_x^2 \vec{A} - \frac{1}{c^2} \partial_t^2 \vec{A} = - \frac{4\pi}{c} \vec{J}_T \tag{5.1}$$

where $\vec{J}_T$ is the transverse current. From here on it is assumed that $\vec{A}$ is in the $\hat{y}$ direction and that the wave vectors $\vec{k}_i$ $(i=1,2)$ are in the $\hat{x}$ direction so that vector notation can be dropped.

Neglecting higher order terms, A and J_T can be expanded as:

$$A = A_1(x,t)\, e^{i\xi_1} + A_2(x,t)\, e^{i\xi_2} + c.c,$$

$$J_T = J_1(x,t)\, e^{i\xi_1} + J_2(x,t)\, e^{i\xi_2} + c.c. \tag{5.2}$$

where n_o is the equilibrium density and $\xi_1 = (k_1 x - \omega_1 t)$ and $\xi_2 = (k_2 x - \omega_2 t)$. It is also assumed that electron density can be expanded as:

$$n_e = n_o + (\tilde{n}_e(x,t)\, e^{i\xi_3} + c.c.) \tag{5.3}$$

where $\xi_3 = \xi_1 - \xi_2$. Terms of the form $\exp[i(\xi_1 + \xi_2)]$ have been neglected since they are off resonant.

The transverse current can be found from the transverse electron fluid velocity u.

$$J_T = - n_e e u \tag{5.4}$$

The ion contribution to the transverse current has been neglected since it is smaller than the electron contribution by a factor of (m_e/m_i). The transverse electron fluid velocity u is expanded as:

$$u = u_1(x,t)\, e^{i\xi_1} + u_2(x,t)\, e^{i\xi_2} + \text{c.c.} \tag{5.5}$$

The electron fluid velocity can be derived from:

$$\partial_t u = - \nu_{ei} u - \frac{e}{m_e} E \tag{5.6}$$

where ν_{ei} is the electron-ion collision frequency and $E = -\frac{1}{c}\partial_t A$ is the electric field due to the electromagnetic beams. Magnetic terms are of order (u/c) and have been neglected. Equation (5.6) can be split into two parts and written as:

$$\partial_t (u_i\, e^{i\xi_i}) = - \nu_{ei} u_i\, e^{i\xi_i} + \frac{e}{m_e c} \partial_t (A_i e^{i\xi_i}) \tag{5.7}$$

or

$$(\partial_t - i\omega_i)\, u_i = - \nu_{ei} u_i + \frac{e}{m_e c} (\partial_t - i\omega_i)\, A_i$$

for $i = 1, 2$.

If $\partial_t u_i \ll \omega_i u_i$ the above equation can be approximated as:

$$-i\omega_i u_i = -\nu_{ei} u_i - i\omega_i \frac{e}{m_e c} A_i$$

or

$$u_i = \eta_i \frac{eA_i}{m_e c} \tag{5.8}$$

where

$$\eta_i = \frac{\omega_i^2 - i\nu_{ei}\omega_i}{\omega_i^2 + \nu_{ei}^2} \simeq 1 - \frac{i\nu_{ei}}{\omega_i} \tag{5.9}$$

where it is assumed that $(\nu_{ei}/\omega_i)^2 \ll 1$.

From (5.3) and (5.4) the transverse component of the current may be shown to be:

$$J_T = - eu[n_o + (\tilde{n}_e e^{i\xi_3} + c.c.)] \tag{5.10}$$

Combining (5.8) and (5.10) the resonant current terms are found to be:

$$J_1(x,t) = - \frac{\eta_1}{4\pi c} \omega_{pe}^2 A_1 - \frac{\eta_2}{4\pi c} \omega_{pe}^2 A_2 \left(\frac{\tilde{n}_e}{n_o}\right)$$

$$J_2(x,t) = - \frac{\eta_2}{4\pi c} \omega_{pe}^2 A_2 - \frac{\eta_1}{4\pi c} \omega_{pe}^2 A_1 \left(\frac{\tilde{n}_e^*}{n_o}\right) \tag{5.11}$$

Equation (5.1) is split into two parts and its harmonic dependence is inserted. Then:

$$\partial_t^2 \to (\partial_t - i\omega_i)^2 = \partial_t^2 - \omega_i^2 - 2i\omega_i \partial_t$$

$$\partial_x^2 \to (\partial_x - ik_i)^2 = \partial_x^2 - k_i^2 - 2ik_i \partial_x$$

It is assumed that $\partial_t^2 A_1 \ll \omega_1^2 A_1$ and $\partial_x^2 A_1 \ll k_1^2 A_1$ and that small terms can be neglected. Then:

$$(\partial_t^2 - c^2\partial_x^2) + (-\omega_1^2 + k_1^2 c^2 - 2i\omega_1\partial_t - 2ic^2 k_1 \partial_x) \tag{5.12}$$

By using this relation the component of equation (5.1) with an $e^{i\xi_1}$ dependence can be written as:

$$[(-\omega_1^2 + k_1^2 c^2 + \omega_{pe}^2) - 2i\omega_1\partial_t - 2ic^2 k_1\partial_x - \frac{i\nu_{ei}\omega_{pe}^2}{\omega_1}] A_1$$
$$= -\eta_2 \omega_{pe}^2 A_2 (\frac{\tilde{n}_e}{n_o}) \tag{5.13}$$

It is assumed that the waves in the electromagnetic beams obey the dispersion relation:

$$-\omega_1^2 + k_1^2 c^2 + \omega_{pe}^2 = 0$$

so the first term in (5.13) can be set to zero. Any error in this assumption due to nonlinear frequency shifts does not lead to an error in the solution. Instead it shows up in the complex solution of $A_1(x,t)$ rotating in the complex plane.

Equation (5.13) and the corresponding equation that can be derived from the component of (5.1) with a $e^{i\xi_2}$ dependence can be expressed as:

$$[\partial_t + c_1 \partial_x + \Gamma_1] A_1(x,t) = -(\frac{i}{2}) \frac{\omega_{pe}^2}{\omega_1} \frac{\tilde{n}_e}{n_o} A_2(x,t) \tag{5.14}$$

$$[\partial_t + c_2 \partial_x + \Gamma_2] A_2(x,t) = - (\tfrac{i}{2}) \frac{\omega_{pe}^2}{\omega_2} \frac{\tilde{n}_e^*}{n_o} A_1(x,t)$$

where $c_i = \frac{c^2 k_i}{\omega_i}$ is the group velocity of the electromagnetic beams in the plasmas, $\Gamma_i = \frac{\nu_{ei}\omega_{pe}^2}{2\omega_i^2}$ are the collisional absorption rates for the beams, and * denotes the complex conjugate.

It should be noted that (5.14) makes no assumption on the magnitude of, or rates of change of the density fluctuation $\tilde{n}_e(x,t)$. In fact the only restriction on the validity of equations (5.14) is that the magnitude of $A_i(x,t)$ does not change significantly in one period or wavelength of the electromagnetic beams.

The Manley-Rowe relations (photon conservation) can be easily derived from equations (5.14). Neglect collisional absorption and multiply the first equation by $\omega_1 A_1^*$ and the second equation by $\omega_2 A_2^*$. Add each of these equations to its complex conjugate and then add the two resulting equations. The result is:

$$\omega_1 [\partial_t + c_1 \partial_x] |A_1|^2 + \omega_2 [\partial_t + c_2 \partial_x] |A_2|^2 = 0 \qquad (5.15)$$

The intensity in each beam can be expressed as:

$$I_i = \frac{\omega_i |k_i|}{2\pi} |A_i|^2 \qquad \left(\frac{\text{ergs}}{\text{sec cm}^2} \right)$$

Substituting this into the above equation yields:

$$[\partial_t + c_1 \partial_x] \left(\frac{I_1}{|k_1|}\right) + [\partial_t + c_2 \partial_x] \left(\frac{I_2}{|k_2|} \right) = 0 \qquad (5.16)$$

Since the photon density in each beam is proportional to (I_i/k_i) this equation implies photon conservation. An equivalent form for (5.16) is (use $k_i = (\omega_i c_i)/c^2$)

$$[\partial_t + c_1 \partial_x] (I_1/ |c_1|\omega_1) + [\partial_t + c_2\partial_x] (I_2/ |c_2|\omega_2) = 0$$

The Manley-Rowe relations can be cast in a simpler form. The energy density of a beam in the plasma is $E_i = I_i/ |c_i|$. Equation (5.16) can be rewritten as:

$$\frac{c_1}{k_1} (\partial_t + c_1\partial_x) E_1 + \frac{c_2}{k_2} (\partial_t + c_2 \partial_x) E_2 = 0$$

If W_1 is defined as the rate at which energy is being transferred out of the high frequency beam and W_2 is the rate at which energy is being transferred into the low frequency beam,

$$W_1 = - (\partial_t + c_1 \partial_x) E_1$$

$$W_2 = + (\partial_t + c_2 \partial_x) E_2$$

Using the above definitions and the fact that $\frac{c_i}{k_i} = \frac{c^2}{\omega_i}$, the above equation can be expressed as,

$$\frac{W_1}{\omega_1} = \frac{W_2}{\omega_2}$$

If the rate at which energy is transferred to the beat frequency electrostatic wave is defined as W_3, the conservation of energy principle implies that:

$$W_3 = W_1 - W_2 = W_1(1 - \omega_2/\omega_1) = \frac{\omega_3}{\omega_1} W_1$$

Combining the above two equations yields:

$$\frac{W_1}{\omega_1} = \frac{W_2}{\omega_2} = \frac{W_3}{\omega_3} \tag{5.17}$$

5.2 Density Fluctuations in the Presence of Antiparallel Electromagnetic Beams

Equations (5.14) can be used to describe the coupling of two antiparallel electromagnetic beams in the presence of an electron density fluctuation. The ponderomotive force from these two beams, however drives a density fluctuation of frequency ω_3 and wave-vector $\vec{k}_3$ where $\omega_3 = \omega_1 - \omega_2$ and $\vec{k}_3 = \vec{k}_1 - \vec{k}_2$. In this section a general relation for the electron density fluctuation resulting from the mixing of two antiparallel electromagnetic beams is derived. The effects of ion mobility and electron-ion collisions are included in this analysis.

In this treatment the following four forces acting on electrons and ions are accounted for.

(1) $F_{c\epsilon}$ - Force experienced by species ϵ (ϵ = e or i for electrons or ions) due to the coulomb electric field resulting from electron and ion density fluctuations.

(2) [illegible] - Ponderomotive force on electrons.

(3) F_i - Ponderomotive force on ions.

(4) F_{col} - Phenomenological force on electrons to account for electron-ion collisions. An equal and opposite force is applied to ions.

In addition to these four forces, we also define:

(1) $F_{Te} = F_{ce} + F_e + F_{col}$ - Total force acting on electrons.

(2) $F_{Ti} = F_{ci} + F_i - F_{col}$ - Total force acting on ions.

The double Fourier transform of all quantities $Q(x,t)$

that vary as $\exp[i(k_3x - \omega_3t)]$ is taken. The transform has been defined symmetrically as:

$$Q(k,\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} Q(x,t)\, e^{-i(kx-\omega t)}\, dt\, dx$$

$$Q(x,t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} Q(k,\omega)\, e^{i(kx-\omega t)}\, dk\, d\omega$$

With this definition taking the transform of a differential equation in x and t means that:

$$\partial_t \rightarrow -i\omega \quad \text{and} \quad \partial_x \rightarrow ik$$

The coulomb force on species ε can be written as:

$$F_{c\varepsilon}(x,t) = q_\varepsilon\{E_{ce}(x,t) + E_{ci}(x,t)\} \tag{5.18}$$

where q_ε is the charge of species ε, and $E_{c\varepsilon}$ is the component of the coulomb field due to the density fluctuations of species ε.

The ponderomotive force on species ε has been derived in Appendix C as:

$$F_\varepsilon(x,t) = -\frac{iq_\varepsilon^2}{m_\varepsilon c^2}\, k_3A_1(x,t)A_2^*(x,t)\exp[i(k_3x - \omega_3t)] + \text{c.c.} \tag{5.19}$$

where m_ε is the particle mass of species ε and c = speed of light.

$F_{col}(k,\omega)$ can be expressed as a function of $E_{ce}(k,\omega)$ and $E_{ci}(k,\omega)$. The average force on electrons due to electron-ion collisions can be expressed as:

$$F_{col} = m_e(v_i - v_e) \cdot \nu_{ei} \tag{5.20}$$

where v_ε is the average velocity of particles of species ε, and ν_{ei} is the electron-ion collision frequency. However from the relations:

$$v_\varepsilon = \frac{J_\varepsilon}{n_o q_\varepsilon} \;\; ; \;\; \nabla \cdot E_{c\varepsilon}(x,t) = 4\pi\, \rho_\varepsilon(x,t) \;\; ; \text{ and}$$

$\frac{\partial \rho_\varepsilon}{\partial t} + \nabla \cdot J_\varepsilon = 0$ where J_ε is the current due to species ε, and ρ_ε is the charge density due to perturbations in the number density of species ε it is found that

$$v_\varepsilon(k,\omega) = \frac{i\omega E_{c\varepsilon}(k,\omega)}{4\pi n_o q_\varepsilon} \tag{5.21}$$

Combining (5.20) and (5.21) yields:

$$F_{col}(k,\omega) = \frac{i m_e \omega\, \nu_{ei}}{4\pi n_o e} \{E_{ci}(k,\omega) + E_{ce}(k,\omega)\}$$

or

$$F_{col}(k,\omega) = ie \frac{\omega \nu_{ei}}{\omega_{pe}^2} \{E_{ci}(k,\omega) + E_{ce}(k,\omega)\} \tag{5.22}$$

where $\omega_{pe} = \left(\frac{4\pi n_o e^2}{m_e}\right)^{1/2}$ is the electron plasma frequency. Combining equations (5.18) and (5.22):

$$F_{col}(k,\omega) = -\frac{i\omega\nu_{ei}}{\omega_{pe}^2} F_{ce}(k,\omega) \tag{5.23}$$

To simplify algebra the following constants are defined:

$$a \equiv \frac{i\omega\nu_{ei}}{\omega_{pe}^2} \qquad\qquad b \equiv 1 - a$$

Then from the definitions of F_{Te} and F_{Ti}:

$$F_{Te} = [F_e + b F_{ce}]$$

$$F_{Ti} = [F_i - b F_{ce}] \qquad (5.24)$$

Using a kinetic approach, and solving the linearized Vlasov equation (see Appendix B) it is found that the coulomb field resulting from fluctuations of species α can be expressed as:

$$E_{c\varepsilon}(\omega,k) = -\frac{1}{q_\varepsilon}\chi_\varepsilon(\omega,k)\ F_{T\varepsilon}(\omega,k) \qquad (5.25)$$

where χ_ε is the linear susceptability of species ε. From Appendix B:

$$\chi_\varepsilon(\omega,k) = 2\frac{\omega_{p\varepsilon}^2}{k^2}\frac{1}{v_{\theta\varepsilon}^2}\{1 + i\,\alpha_{o\varepsilon}F_o^{\ \varepsilon}\} \qquad (5.26)$$

where

$$\omega_{p\varepsilon}^{\ 2} = \frac{4\pi n_o e^2}{m_\varepsilon}\ ,\quad v_{\theta\varepsilon} = \left(\frac{2k_B T_\varepsilon}{m_\varepsilon}\right)^{1/2},$$

$$\alpha_{o\varepsilon} = \frac{\omega}{kv_{\theta\varepsilon}}\ ,\quad F_o^{\ \varepsilon} = \frac{ik}{\sqrt{\pi}}\int_{-\infty}^{\infty}\frac{e^{-v^2/v_{\theta\varepsilon}^{\ 2}}}{(\omega_3 - k_3 v)}\,dv,$$

and T_ε is the temperature of species ε. Techniques of evaluating $F_o^{\ \varepsilon}$ are discussed in Stix[31] (Chapter 8). From equations (5.18), (5.24), and (5.25):

$$E_{ce}(k,\omega) = -\chi_e\{-\frac{F_e}{e} + b(E_{ce} + E_{ci})\}$$

$$E_{ci}(k,\omega) = -\chi_i\{\frac{F_i}{e} + b(E_{ce} + E_{ci})\} \qquad (5.27)$$

Solving equations (5.27) for E_{ce} and E_{ci} yields:

$$E_{ce} = + \frac{\chi_e \{F_e (1 + b\,\chi_i) + b\,\chi_i\,F_i\}}{e(1 + b\,\chi_e + b\,\chi_i)}$$

$$E_{ci} = - \frac{\chi_i \{F_i (1 + b\,\chi_e) + b\,\chi_e\,F_e\}}{e(1 + b\,\chi_e + b\,\chi_i)} \qquad (5.28)$$

Writing the total species density $n_{T\varepsilon}(x,t)$ as:

$$n_{T\varepsilon}(x,t) = n_o + n_\varepsilon(x,t)$$

and using,

$$\nabla \cdot E_{c\varepsilon}(x,t) = 4\pi\, q_\varepsilon\, n_\varepsilon(x,t)$$

yields,

$$n_\varepsilon(k,\omega) = \frac{ik\, E_{c\varepsilon}(k,\omega)}{4\pi\, q_\varepsilon} \qquad (5.29)$$

Combining (5.28) and (5.29) and ignoring F_i since $F_i \ll F_e$ yields:

$$n_e(k,\omega) = - \frac{ik}{4\pi e^2} \frac{\chi_e\, F_e (1 + b\,\chi_i)}{(1 + b\,\chi_e + b\,\chi_i)} \qquad (5.30)$$

The Fourier transform of $F_e(x,t)$ defined in equation (5.19) may be shown to be:

$$F_e(k,\omega) = - \frac{ie^2 k_3}{m_e c^2} f(k,\omega) + \frac{ie^2 k_3}{m_e c^2} g(k,\omega)$$

where

$$f(k,\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A_1(x,t)\, A_2^*(x,t)\, e^{i(k_3 x - \omega_3 t)}\, e^{-i(kx - \omega t)} dx\, dt$$

(5.31)

and

$$g(-k,-\omega) = f^*(k,\omega)$$

If $D(k,\omega)$ is defined by:

$$D(k,\omega) = -\frac{kk_3}{4\pi m_e c^2} \frac{\chi_e(1 + b\,\chi_i)}{(1 + b\,\chi_i + b\,\chi_e)} \tag{5.32}$$

it may be shown that $D(-k,-\omega) = -D^*(k,\omega)$. If the inverse Fourier transform of equation (5.30) is now taken and use made of equations (5.31) and (5.32), the electron density perturbation may be written as:

$$n_e(x,t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} D(k,\omega)\, f(k,\omega)\, e^{i(kx-\omega t)} dk\, d\omega + c.c. \tag{5.33}$$

The expression for $\tilde{n}_e(x,t)$ as required for equation (5.14) may be written as:

$$\tilde{n}_e(x,t) = e^{-i(k_3x-\omega_3t)} \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} D(k,\omega)\, f(k,\omega) e^{i(kx-\omega t)} dk\, d\omega \tag{5.34}$$

5.3 Case of Slowly Varying Electromagnetic Beam Amplitudes

If both $A_1(x,t)$ and $A_2(x,t)$ vary slowly on a time scale defined by the damping rate of the beat wave and on a length scale defined by the wavelength of the beat wave, equation (5.34) can be approximated. Under these conditions $f(k,\omega)$ is sharply resonant and has significant values only for $k \approx + k_3$ and $\omega \approx + \omega_3$. Equation (5.34) can be approximated as:

$$\tilde{n}_e(x,t) = e^{-i(k_3 x - \omega_3 t)} \, D(k_3,\omega_3) \, f(x,t)$$

where $f(x,t)$ is the double Fourier transform of $f(k,\omega)$ defined in equation (5.31). Using this definition:

$$\tilde{n}_e(x,t) = D(k_3,\omega_3) \, A_1(x,t) \, A_2^*(x,t) \tag{5.35}$$

Equations (5.14) and (5.35) form a set of three coupled equations that describe the beat-frequency mixing process for slowly varying A_1 and A_2.

By inserting (5.35) into (5.14) it is found that:

$$\begin{aligned} [\partial_t + c_1\partial_x + \Gamma_1]A_1(x,t) &= -\left(\frac{1}{2}\right) \frac{\omega_{pe}^2}{\omega_1} \cdot \frac{1}{n_o} D(k_3,\omega_3)A_1A_2A_2^* \\ [\partial_t + c_2\partial_x + \Gamma_2]A_2(x,t) &= -\left(\frac{1}{2}\right) \frac{\omega_{pe}^2}{\omega_2} \cdot \frac{1}{n_o} D^*(k_3,\omega_3)A_1A_1^*A_2 \end{aligned} \tag{5.36}$$

Multiplying the top equation by A_1^* and the bottom equation by A_2^*, and adding each to its complex conjugate yields:

$$[\partial_t + c_1\partial_x + 2\Gamma_1]\ |A_1(x,t)|^2 = \frac{i}{\omega_1}(\beta - \beta^*)\ |A_1|^2\ |A_2|^2$$

$$[\partial_t + c_2\partial_x + 2\Gamma_2]\ |A_2(x,t)|^2 = -\frac{i}{\omega_2}(\beta - \beta^*)|A_1|^2\ |A_2|^2 \qquad (5.37)$$

where,

$$\beta = -\frac{1}{2}\omega_{pe}^{\ 2} \cdot \frac{1}{n_o} D(k_3,\omega_3)$$

Now using:

$$|A_i|^2 = \frac{2\pi}{\omega_i|k_i|}\ I_i \qquad (5.38)$$

where I_i is the beam intensity;

$$[\partial_t + c_1\partial_x + 2\Gamma_1]\ I_1 = \frac{2\pi i}{\omega_1\omega_2|k_2|}\ (\beta - \beta^*)\ I_1 I_2$$

$$[\partial_t + c_2\partial_x + 2\Gamma_2]\ I_2 = -\frac{2\pi i}{\omega_1\omega_2|k_1|}\ (\beta - \beta^*)\ I_1 I_2$$

or if,

$$\alpha \equiv \frac{2\pi i}{\omega_1\omega_2|k_2|c_1}\ (\beta - \beta^*) \qquad (5.39)$$

$$[\partial_t + c_1\partial_x + 2\Gamma_1]\ I_1 = \alpha c_1\ I_1 I_2$$

$$[\partial_t + c_2\partial_x + 2\Gamma_2]\ I_2 = -\left|\frac{k_2}{k_1}\right|\alpha c_1\ I_1 I_2 \qquad (5.40)$$

The coupling coefficient α can be rewritten as:

$$\alpha = \frac{2\pi\omega_{pe}^{2}}{n_o\omega_1\omega_2 k_2 c_1} \text{ Im } \{D(k_3,\omega_3)\} \tag{5.41}$$

The parameter α must be multiplied by 10^{-7} if I_i is expressed in W/cm^2.

The rate that energy is deposited in the plasma due to beat frequency mixing, W_3, can be calculated from the conservation of energy principle through the use of equations (5.40) and (5.41). The deposited energy W_3 is set equal to the difference in the transfer rate of energy from the first beam and the transfer rate of energy into the second beam. The energy density of a beam in the plasma is $\mathcal{E}_i = I_i/|c_i|$ ergs/cm^3. Neglecting collisional absorption of the beams, equations (5.40) can be rewritten as:

$$(\partial_t + c_1\partial_x)\ \mathcal{E}_1 = \alpha\ I_1 I_2$$

$$(\partial_t + c_2\partial_x)\ \mathcal{E}_2 = -\left|\frac{k_2}{k_1}\right| \left|\frac{c_1}{c_2}\right| \alpha\ I_1 I_2$$

The left hand side of these equations is clearly the rate of energy transfer into the beams. Thus, from the definition of W_3:

$$W_3 = -\left[(\partial_t + c_1\partial_x)\ \mathcal{E}_1 + (\partial_t + c_2\partial_x)\ \mathcal{E}_2\right]$$

or

$$W_3 = -\alpha\ I_1 I_2 \left(1 - \left|\frac{k_2}{k_1}\right| \left|\frac{c_1}{c_2}\right|\right)$$

Using $c_i = \dfrac{c^2 k_i}{\omega_i}$, $\left|\dfrac{k_2}{k_1}\right| \left|\dfrac{c_1}{c_2}\right|$ can be rewritten as $\dfrac{\omega_2}{\omega_1}$. Then:

$$W_3 = -\frac{\omega_3}{\omega_1}\ \alpha\ I_1 I_2 = -\frac{\omega_3 c_1}{c^2 k_1}\ \alpha\ I_1 I_2$$

Using the definition of α from equation (5.41) this expression becomes:

$$W_3 = \frac{-2\ \pi\ \omega_{pe}^2\ \omega_3}{n_o c^2 \omega_1 \omega_2 k_1 k_2}\ \mathrm{Im}\{D(k_3,\omega_3)\}\ I_1(x,t)\ I_2(x,t) \tag{5.42}$$

It is interesting to note that equation (5.42) can also be derived by the more direct method of setting the plasma heating rate to the ponderomotive force on the electrons times the electron particle flux. The electron particle flux I_e can be found from the continuity equation:

$$\frac{\partial}{\partial t} n_e(x,t) = -\frac{\partial}{\partial x} I_e(x,t) \tag{5.43}$$

and equation (5.30):

$$I_e(k,\omega) = -\frac{i\omega}{4\pi e^2}\ \frac{\chi_e(1 + b\ \chi_i)}{1 + b\ \chi_e + b\ \chi_i}\ F_e(k,\omega)$$

or, using (5.32):

$$I_e(k,\omega) = \frac{i\omega m_e c^2}{k\ k_3 e^2}\ D(k,\omega)\ F_e(k,\omega) \tag{5.44}$$

For slowly varying $F_e(x,t)$, $F_e(k,\omega)$ is sharply resonant in k and ω, and the inverse transform of (5.44) can be approximated so:

$$I_e(x,t) = \tilde{I}_e(x,t)\ \exp[i(k_3 x - \omega_3 t)]\ + \text{c.c.} \tag{5.45}$$

where,

$$\tilde{I}_e(x,t) = \frac{i\omega_3 m_e c^2}{k_3^2 e^2} D(k_3,\omega_3) \tilde{F}_e(x,t) \qquad (5.46)$$

and,

$$\tilde{F}_e(x,t) = -\frac{ie^2}{m_e c^2} k_3 A_1(x,t) A_2^*(x,t) \qquad (5.47)$$

The rate that energy is deposited in the plasma, W_3, is set to the ponderomotive force on the electrons times the electron particle flux.

$$W_3 = 2 \operatorname{Re}\{\tilde{I}_e \tilde{F}_e^*\} \qquad (5.48)$$

Using equations (5.46) and (5.47) in equation (5.48) yields:

$$W_3 = 2 \operatorname{Re}\{ \frac{i\omega_3 e^2}{m_e c^2} D(k_3,\omega_3) A_1 A_1^* A_2 A_2^* \}$$

Using equation (5.38)

$$W_3 = 2 \operatorname{Re}\{ \frac{4\pi^2 i\omega_3 e^2}{m_e c^2 \omega_1 \omega_2 k_1 k_2} D(k_3,\omega_3)\} I_1(x,t) I_2(x,t)$$

or

$$W_3 = -\frac{2\pi\omega_{pe}^2 \omega_3}{n_o c^2 \omega_1 \omega_2 k_1 k_2} \operatorname{Im}\{D(k_3,\omega_3)\} I_1(x,t) I_2(x,t)$$

which is identical to (5.42).

It is interesting to compare the results of this derivation with the results of Capjack and James[30] where a different type of derivation was used. For this comparison collisional effects are neglected since they are accounted for in a different manner in the two treatments. The present work accounts for electron-ion collisions in a more consistent manner. First, the beat frequency electric field $E_3 \equiv (E_{ce} + E_{ci})$ is found. From equations (5.28):

$$E_3(\omega,k) = \frac{\chi_e}{1 + \chi_i + \chi_e} \quad \frac{F_e}{e} \tag{5.49}$$

where b has been set to 1 and F_i has been set to zero. Again, if $F_e(\omega,k)$ is sharply resonant around $\omega = \pm\,\omega_3$ and $k = \pm\, k_3$ the inverse Fourier transform of (5.49) can be approximated.

$$E_3(x,t) = \tilde{E}_3(x,t)\ \exp[i(k_3x - \omega_3 t)] + c.c. \tag{5.50}$$

where

$$\tilde{E}_3(x,t) = \frac{\chi_e'}{1 + \chi_2' + \chi_e'} \quad \frac{\tilde{F}_e(x,t)}{e} \tag{5.51}$$

$$\chi_\alpha' = \chi_\alpha(\omega_3, k_3)$$

and $\tilde{F}_e(x,t)$ is defined in (5.47).

$\tilde{F}_e(x,t)$ must be cast in a different form to compare with the results of Capjack and James. Assuming $\omega_i >> \omega_{pe}$ or $\omega_i = ck_i$, equation (5.38) can be approximated as:

$$|A_i| = \left(\frac{2\pi c}{\omega_i{}^2} \right)^{1/2} \sqrt{I_i}$$

Then:

$$A_1 A_2^* = \frac{2\pi c}{\omega_1 \omega_2} \sqrt{I_1 I_2} \ e^{i\phi}$$

where ϕ is a phase factor which can be neglected in this treatment.

Substitution of the above equation into equation (5.47) yields:

$$\tilde{F}_e(x,t) = - i \frac{2\pi c e^2}{m_e c^2} \quad \frac{k_3}{\omega_1\omega_2} [I_1(x,t)\ I_2(x,t)]^{1/2}$$

Again the assumption that $\omega_i >> \omega_{pe}$ or $\omega_i = c\ k_i$ is made and $(\omega_1\omega_2)^{-1}$ is approximated as:

$$(\omega_1\omega_2)^{-1} = \frac{\lambda_1\lambda_2}{(2\pi c)^2}$$

Then:

$$\tilde{F}_e(x,t) = - \frac{ie^2}{2\pi c^3 m_e} \ k_3\ \lambda_1\lambda_2 \sqrt{I_1 I_2}$$

or:

$$\tilde{F}_e(x,t)/e = - i(3.11 \times 10^{-16})\ k_3\ \lambda_1\lambda_2 \sqrt{I_1 I_2} \qquad (5.52)$$

where λ_i is expressed in μm and I_i is expressed in W/cm^2. From (5.26):

$$\chi_s' = 2 \ \frac{\omega_{p\epsilon}^2}{k_3^2} \ \frac{1}{v_{\theta\epsilon}^2} \ \{1 + i\ \alpha_o^\epsilon\ F_o^\epsilon\}$$

After some algebra:

$$\frac{\chi_e'}{1 + \chi_e' + \chi_i'} = \frac{1 + i\alpha_o^e\ F_o^e}{[K^2 + \{1 + i\alpha_o^e\ F_o^e\} + \frac{T_e}{T_i}\{1 + i\alpha_o^i\ F_o^i\}]} \qquad (5.53)$$

where $K \equiv k_3\lambda_D$ and $\lambda_D = v_{\theta_e}/\sqrt{2}\omega_{pe}$ is the Debye wavelength. From equations (5.51), (5.52), and (5.53):

$$\tilde{E}_3(x,t) = \frac{i(3.11 \times 10^{-16}) k_3\ \lambda_1\lambda_2(1 + i\alpha_{oe}F_o^e) \sqrt{P_1P_2}}{[K^2 + \{1 + i\alpha_o^e\ F_o^e\} + \frac{T_e}{T_i}\{1 + i\alpha_o^i\ F_\rho^i\}]} \qquad (5.54)$$

For $T_e = T_i$ this is in exact agreement with the value for the coulomb field found by Capjack and James[30].

The plasma heating rate W_3 can also be cast into a form similar to that of Capjack and James[30]. It is most convenient to express $\tilde{I}_e(x,t)$ and $\tilde{F}_e(x,t)$ in terms of $\tilde{E}_{ce}(x,t)$ before using (5.48) to find W_3. By using the continuity equation:

$$\partial_t \, n_e(x,t) = - \partial_x \, I_e(x,t)$$

and Gauss' Law:

$$\partial_x \, E_{ce}(x,t) = - 4\pi \, e \, n_e$$

it is found that:

$$I_e(k,\omega) = - \frac{i\omega}{4\pi e} \, E_{ce}(k,\omega) \qquad (5.55)$$

Taking the inverse Fourier transform yields:

$$\tilde{I}_e(x,t) = - \frac{i\omega_3}{4\pi e} \, \tilde{E}_{ce}(x,t)$$

where $\tilde{I}_e(x,t)$ is defined in (5.45).

Letting b=1 and χ_i=0 in equation (5.28) and approximating the inverse Fourier transform yields:

$$\tilde{F}_e(x,t) = \frac{\chi_e' + 1}{\chi_e'} \, e \, \tilde{E}_{ce}(x,t) \qquad (5.56)$$

where $\chi_e' \equiv \chi_e(k_3,\omega_3)$. Substituting (5.55) and (5.56) into (5.48) yields:

$$W_3 = \frac{1}{2\pi} \; \mathcal{Im}\{\omega_3 \; \chi_e'\} \; \frac{|\tilde{E}_{ce}|^2}{|\chi_e'|^2}$$

Substitution of values for χ_e' yields:

$$W_3 = \frac{\omega_{pe}^2 \; \alpha_{oe}^2}{k_3 v_{\theta e} \pi} \; \sqrt{\pi} \; \text{sgn} \; (k_3) \; e^{-\alpha_{oe}^2} \; \frac{|E_{ce}|^2}{|\chi_e'|^2} \tag{5.57}$$

The heating rate given by (5.57) is smaller by a factor of $|\chi_e'|^{-2}$ from that derived by Capjack and James[30]. For a cold plasma ($\alpha_{oe} << 1$) the two results differ by a factor of $(\omega_3/\omega_{pe})^4$. The discrepancy can be attributed to an error in the definition of the heating rate used by Capjack and James[30].

5.4 Transient Analysis

In cases where the amplitude of one of the electromagnetic waves changes rapidly in time the quasi-static approximation breaks down. In this section, equations (5.14) and (5.34) are cast in a form such that they can easily be solved numerically.

$$[\partial_t + c_1\partial_x + \Gamma_1]\, A_1(x,t) = -\left(\tfrac{1}{2}\right)\frac{\omega_{pe}^2}{\omega_1}\,\frac{\tilde{n}_e(x,t)}{n_o}\,A_2(x,t)$$

$$[\partial_t + c_2\partial_x + \Gamma_2]\, A_2(x,t) = -\left(\tfrac{1}{2}\right)\frac{\omega_{pe}^2}{\omega_2}\,\frac{\tilde{n}_e^*(x,t)}{n_o}\,A_1(x,t) \qquad (5.14)$$

$$\tilde{n}_e(x,t) = e^{-i(k_3x-\omega_3t)}\cdot\frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{\infty}^{\infty} D(k,\omega)f(k,\omega)e^{i(kx-\omega t)}dk\,d\omega \qquad (5.34)$$

where:

$$f(k,\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} A_1(x,t)A_2^*(x,t)e^{i(k_3x-\omega t)}e^{-i(kx-\omega t)}dx\,dt$$

and

$$D(k,\omega) = -\frac{k\,k_3}{4\pi m_e c^2}\;\frac{\chi_e(1 + b\,\chi_i)}{(1 + b\chi_e + b\chi_i} \qquad (5.32)$$

where χ_e and χ_i are the usual electron and ion linear susceptibilities as calculated from the linear Vlasov equations. The validity of these equations is dependent on the following inequalities.

$$|\partial_t\, A_i(x,t)| \ll |\omega_i \cdot A_i(x,t)|$$

$$i=1,2$$

$$|\partial_x\, A_i(x,t)| \ll |k_i \cdot A_i(x,t)|$$

Since for the mixing of antiparallel beams $|k_3| = |k_1| + |k_2|$, the second of the above inequalities implies that in the region of validity of the above equations $f(k,\omega)$ will be sharply resonant in k, and has significant values only for k near $\pm k_3$. The function $f(k,\omega)$ is not necessarily sharply resonant in ω.

As an example consider the mixing of a beam of wavelength 10.6μm with a pulse with a wavelength of 10.612μm. For this case the frequency and wavevector of the driven electrostatic wave almost satisfies the dispersion relation of an ion-acoustic wave in a 100eV plasma. The frequency $\omega_3 = 2x10^{11}$ sec^{-1} and the wavevector $k_3 = 1.2x10^4$ cm^{-1} or $T_3 = 3.1x10^{-11}$sec, and $\lambda_3 = 5.3x10^{-4}$cm where T_3 and λ_3 are the period and wavelength of the driven electro-static wave. If the pulse is 10^{-11} seconds or 0.3 cm long,

$$\Delta\ \omega\ \alpha\ \frac{1}{\Delta t} = 10^{11} \rightarrow \Delta\omega/\omega_3 = 0.5$$

$$\Delta\ k\ \alpha\ \frac{1}{\Delta x} = 3.3 \rightarrow \Delta k/k_3 = 2.8x10^{-4}$$

Such a pulse is highly resonant in k but the frequency spread is almost as large as the frequency itself.

When $f(k,\omega)$ is sharply resonant in k around k_3 so that $D(k,\omega)$ does not change much over the range of k, the k integration in (5.34) can be approximated:

$$\tilde{n}_e(x,t) = e^{-i(k_3x-\omega_3t)} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x,\omega)D(k_3,\omega)e^{-i\omega t}\, d\omega \quad (5.58)$$

where

$$f(x,\omega) \equiv \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(k,\omega)\ e^{ikx}\, dk \quad (5.59)$$

By the Convolution Theorem, (5.58) can be rewritten as:

$$\tilde{n}_e(x,t) = e^{-i(k_3x-\omega_3t)} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} f(x,\tau)\, D(k_3,t-\tau) d\tau \quad (5.60)$$

where

$$D(k_3,t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} D(k_3,\omega)\, e^{-i\omega t}\, d\omega \quad (5.61)$$

Now if

$$\tilde{f}(x,t) \equiv f(x,t)\, e^{-ik_3x} = A_1(x,t)\, A_2^*(x,t)\, e^{-i\omega_3t} \quad (5.62)$$

$\tilde{n}_e(x,t)$ can be expressed as:

$$\tilde{n}_e(x,t) = e^{i\omega_3t} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} \tilde{f}(x,\tau)\, D(k_3,t-\tau)\, d\tau \quad (5.63)$$

In principle equations (5.14) and (5.63) can be solved self-consistently to yield an approximate solution to the evolution of two interacting beams.

In order to numerically calculate the solution to these equations for a short pulse, it is most convenient to transform the equations into a coordinate system where the pulse is stationary. The appropriate variables are t' and x' where:

$$\begin{aligned} t' &= t \qquad & \partial_t &\rightarrow \partial_{t'} - c_2\, \partial_{x'} \\ x' &= x-c_2t \qquad & \partial_x &\rightarrow \partial_{x'} \end{aligned} \quad (5.64)$$

Equations (5.14) transform to:

$$[\partial_{t'} + (c_1 - c_2)\partial_{x'} + \Gamma_1]A_1'(x',t') = R_1\tilde{n}_e'(x',t')A_2'(x',t') \quad (5.65)$$

$$[\partial_{t'} + \Gamma_2]A_2'(x',t') = R_2\,\tilde{n}_e^{*\prime}(x',t')A_1'(x',t') \quad (5.66)$$

where

$$R_1 = -\left(\frac{1}{2}\right)\frac{\omega_{pe}^2}{\omega_1 n_o}$$

$$R_2 = -\left(\frac{1}{2}\right)\frac{\omega_{pe}^2}{\omega_2 n_o} \quad (5.67)$$

The variables A_1, A_2, and n_e have been primed to emphasize the fact that they have a new functional form in the new coordinate system.

In terms of the new variables, equation (5.63) can be written as:

$$\tilde{n}_e(x'+c_2t',t') = e^{i\omega_3 t'}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{t'}\tilde{f}(x'+c_2t',\tau)D(k_3,t'-\tau)\,d\tau$$

The functional dependence of, $\tilde{n}_e$ and $\tilde{f}$ is changed in the primed coordinate system so that:

$$\tilde{n}_e'(x',t') = \tilde{n}_e(x' + c_2t',t')$$

$$\tilde{f}'(x',t') = \tilde{f}(x' + c_2t',t') \text{ or } \tilde{f}(x',t') = \tilde{f}'(x' - c_2t',t')$$

Thus,

$$\tilde{f}(x' + c_2t',\tau) = \tilde{f}'(\xi,\tau) = A_1'(\xi,\tau)\,A_2^*(\xi,\tau)\,e^{-i\omega_3 t'} \quad (5.68)$$

where

$$\xi = x' + c_2(t' - \tau)$$

The integral equation for $\tilde{n}_e$ can now be expressed as:

$$\tilde{n}_e'(x',t') = e^{i\omega_3 t'} \frac{1}{\sqrt{2\pi}} \int_{t'+x'/c_2}^{t'} \tilde{f}'(\xi,\tau)\, D(k_3,t'-\tau)\, d\tau \qquad (5.69)$$

The lower limit on the integral comes from assuming $A_2'(x',t') = 0$ for $x' < 0$ and also assuming that $c_2 < 0$ and $c_1 > 0$. Substitution of (5.68) into (5.69) yields:

$$\tilde{n}_e'(x',t') = \frac{1}{\sqrt{2\pi}} \int_{t'+x'/c_2}^{t'} A_1'(\xi,\tau) A_2^{*\prime}(\xi,\tau)\, e^{i\omega_3(t'-\tau)} D(k_3,t'-\tau) d\tau$$

Changing the variable of integration from τ to $\tau' = t' - \tau$ yields:

$$\tilde{n}'(x',t') = \frac{1}{\sqrt{2\pi}} \int_0^{-x'/c_2} A_1'(x'+c_2\tau',t'-\tau') A_2'^{*}(x'+c_2\tau',t'-\tau') \cdot e^{i\omega_3\tau'} D(k_3,\tau')\, d\tau' \qquad (5.70)$$

Equations (5.65), (5.66) and (5.70) form a set of coupled equations from which an approximate solution can be obtained numerically. A numerical solution to these equations will be presented in Chapter 6.

5.5 Results

In this section the dependence of the beat frequency mixing rate (defined through the parameter α) on plasma parameters and the laser difference frequency is displayed graphically. The figures in this chapter were chosen to show the dependence of α on plasma parameters, and to provide numerical data that will be needed in Chapter 6.

Figures 5.1, 5.2, and 5.3 show the beat frequency mixing rate α, the electron density fluctuation level $\tilde{N}_e$, and the ion density fluctuation level $\tilde{N}_i$, as a function of the laser difference frequency ω_3. Here the density fluctuation levels are normalized to be independent of laser beam intensities. That is:

$$\tilde{N}_\varepsilon \equiv \frac{2|\tilde{n}_\varepsilon|}{\sqrt{I_1 I_2}\, n_o}$$

where I_1 and I_2 are expressed in $W \cdot cm^{-2}$. All of these plots show a sharp peak in α, and $\tilde{N}_e$ for ω_3 close to the Langmuir frequency. In Figure 5.1, $T_e >> T_i$ and there is a sharp peak in $\tilde{N}_e$, $\tilde{N}_i$ and α for ω_3 close to the ion-acoustic frequency. It is interesting to note that in Figures 5.2 and 5.3, the plasma density fluctuations are almost independent of ω_3 over a wide range while the parameter α increases monotonically with increasing ω_3. Another interesting feature of these plots is the sharp minimum in $\tilde{N}_e$ and α for ω_3 close to ω_{pi} where $\omega_{pi} = (4\pi n_o e^2/m_i)^{1/2}$ is the ion plasma frequency. The minimum results from the fact that near the ion plasma frequency, the ions move in such a way that their coulomb field cancels the driving field on the electrons.

Figures 5.4, 5.5, and 5.6 show α versus n_o for various difference frequencies, and plasma temperatures. These plots show that at higher temperatures, where Landau damping is strong, α is not nearly as strongly dependent on density as it is at lower temperatures where damping is weaker. Also, for larger difference frequencies, higher temperatures are needed to broaden the resonance.

Figures 5.7, 5.8, and 5.9 show the dependence of α on density and temperature for beat frequency mixing near the ion-acoustic frequency. In these plots it was assumed that $\lambda_1 = 10.247\mu m$ and $\lambda_2 = 10.2605\mu m$. These wavelengths correspond to the R(20) and R(18) transitions in a CO_2 laser respectively. Figure 5.7 shows that α increases monotonically with increasing density over a wide range of densities. Figures 5.8 and 5.9 show α as a function of temperatures. For cases of $T_i << T_e$ sharp peaks are seen in α for ω_3 close to the ion-acoustic frequency.

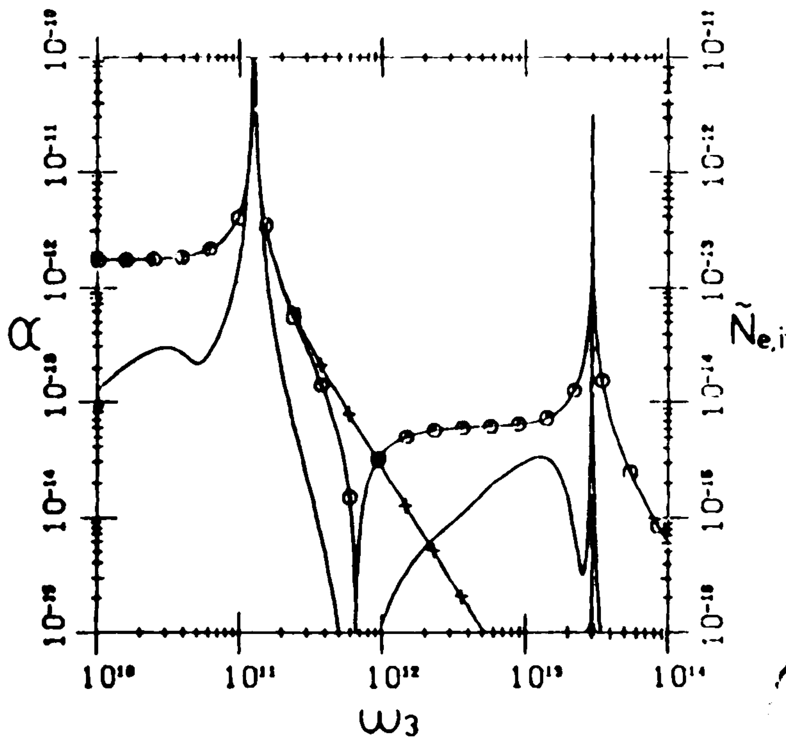

Figure 5.1 α (solid line), $\tilde{N}_e$(o), and $\tilde{N}_i$(+) as a function of ω_3 for n_o=2.5x10^{17} cm^{-3}, T_e=100 eV, T_i=1 ev, and λ_1=10.247μm.

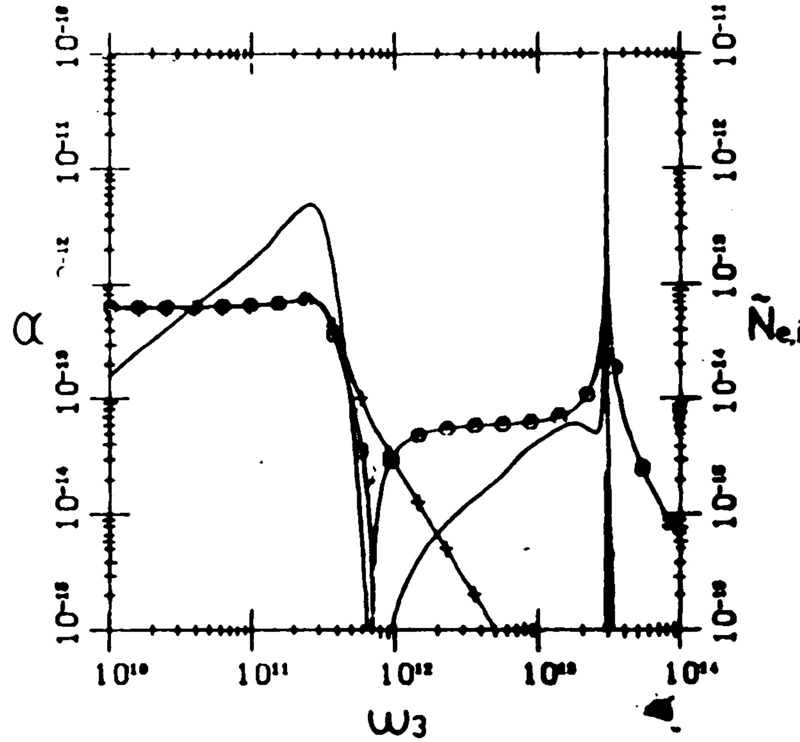

Figure 5.2 α (solid line), $\tilde{N}_e$(o), and $\tilde{N}_i$(+) as a function of ω3 for n_o=2.5x10^{17} cm^{-3}, T_e=160 eV, T_i=120 eV, and λ_1=10.247 μm.

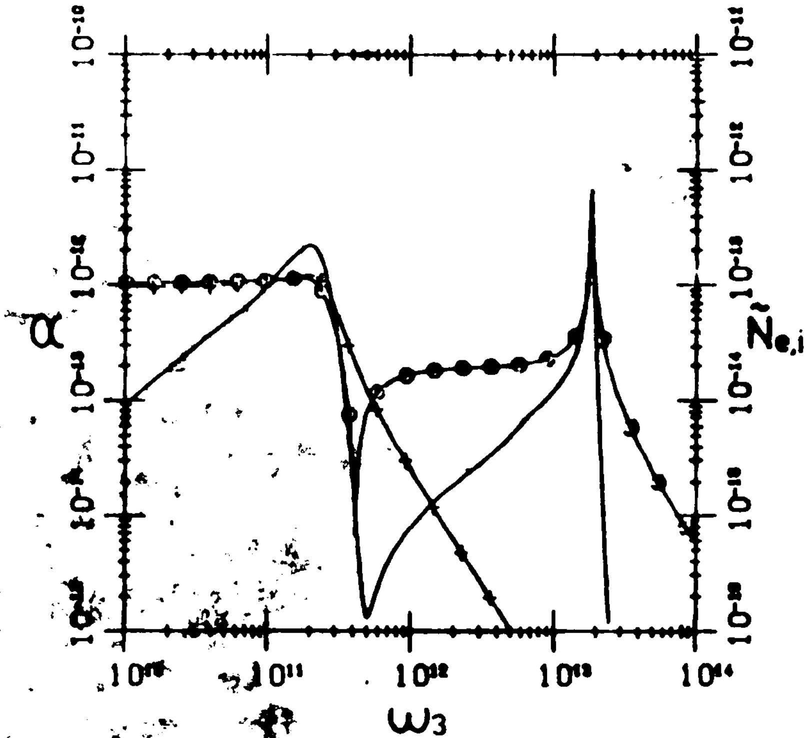

Figure 5.3 α (solid line), $\tilde{N}_e$(o), and $\tilde{N}_i$(+) as a function of ω_3 for n_o=7.5x$10^{16}$$cm^{-3}$, T_e=100 eV, T_i=70 eV and λ_1=9.6 μm.

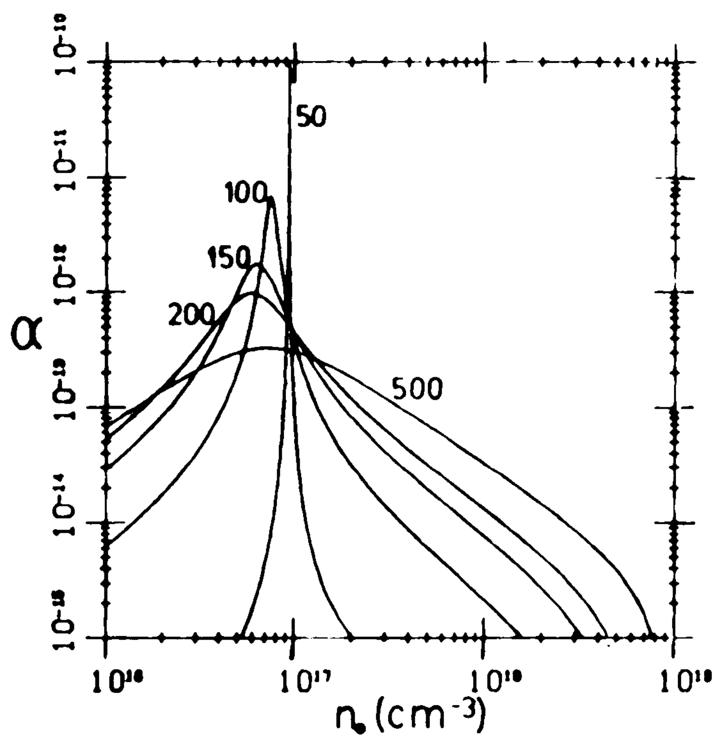

Figure 5.4 α as a function of n_o for λ_1=9.6 μm and λ_2=10.6 μm. The plasma temperature in eV is indicated on the curves.

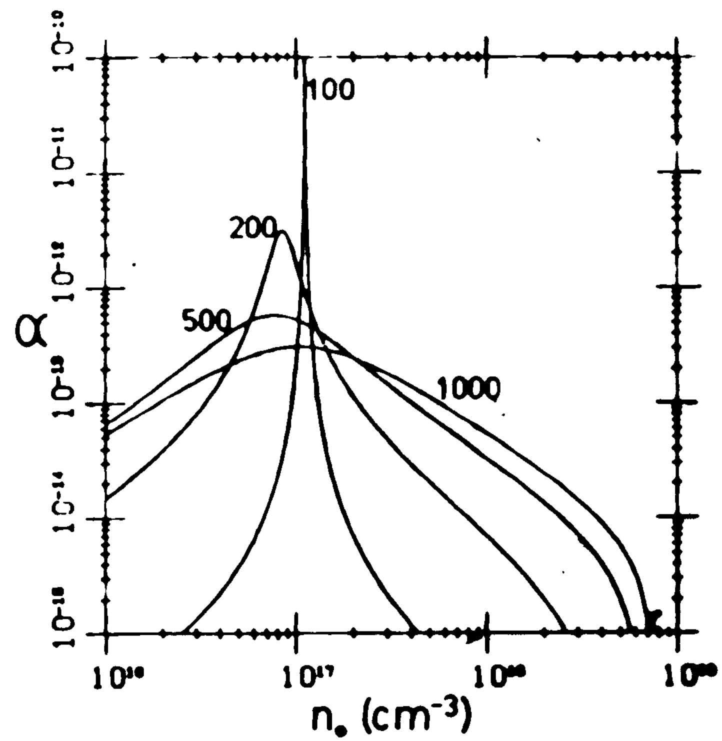

Figure 5.5 α as a function of n_o for λ_1=10.6 μm and λ_2=12.0 μm. The plasma temperature in eV is indicated on the curves.

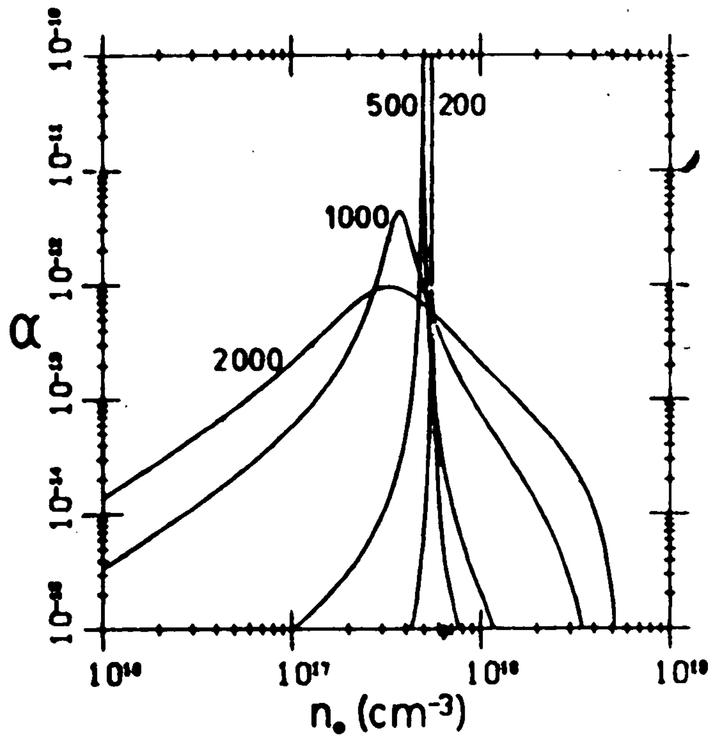

Figure 5.6 α as a function of n_o for λ_1=10.6 μm and λ_2=14.0 μm. The plasma temperature in eV is indicated on the curves.

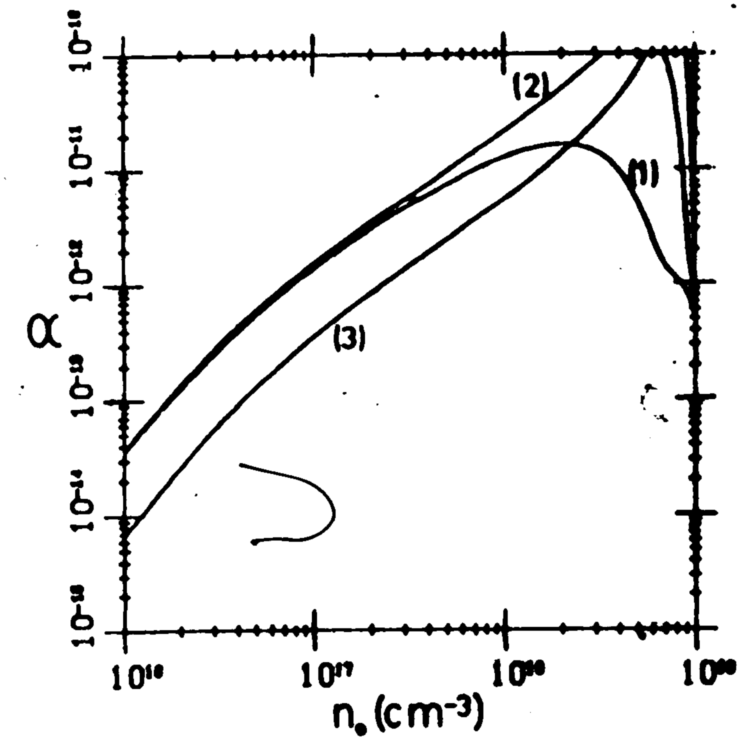

Figure 5.7 α as a function of n_o for λ_1=10.247 μm and λ_2=10.2605 μm. (1) T_e=100eV, T_i=50eV; (2) T_e=200eV, T_i=150eV; (3) T_e=500eV, T_i=300eV.

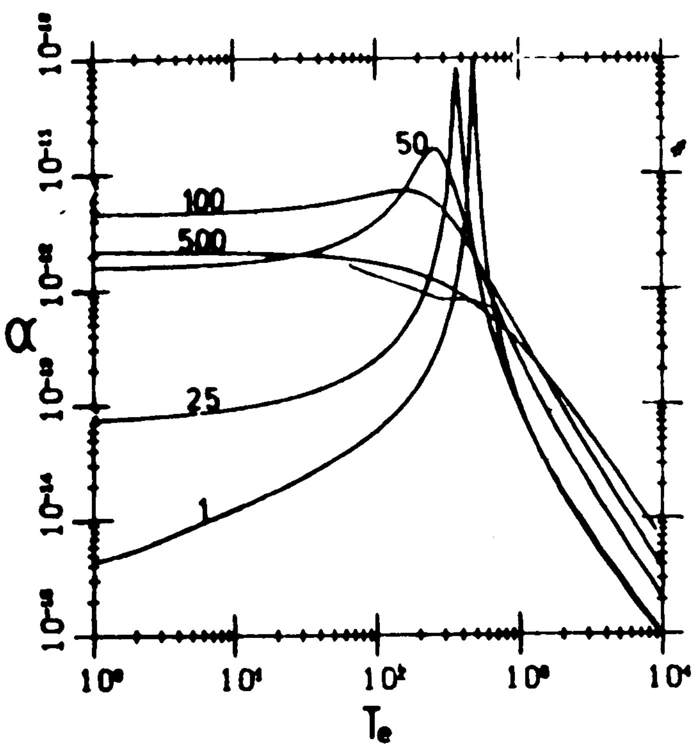

Figure 5.8 α as a function of T_e for n_o=2.5x10^{17} cm^{-3}, λ_1=10.247μm, and λ_2=10.2605μm. The value of T_1 is indicated on the curves.

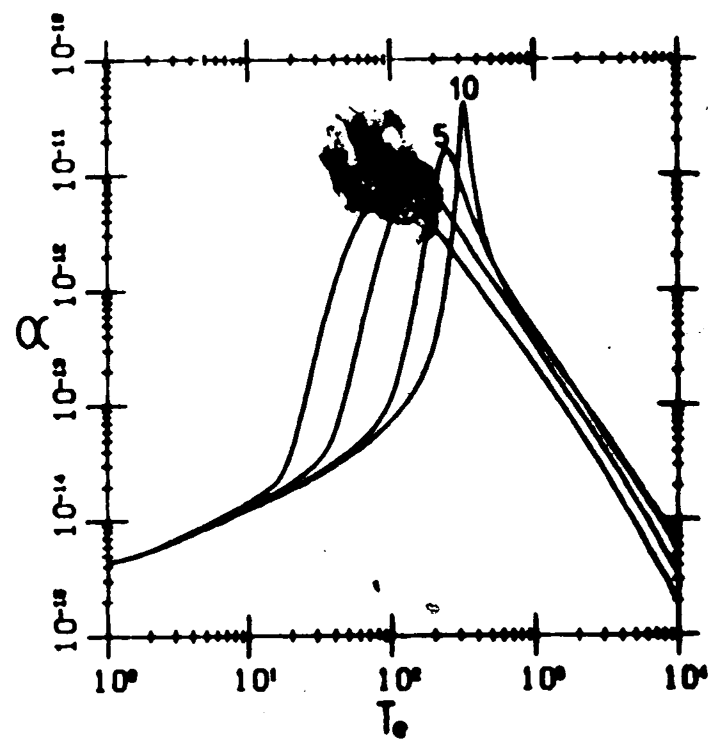

Figure 5.9 α as a function of T_e for n_o=2.5x10^{17} cm^{-3}, λ_1=10.247μm. and λ_2=10.2605μm. Ti is given by (1/a)T_e where values of a are indicated on the curves.

CHAPTER 6

Some Applications of Beat Frequency Mixing

The beat frequency mixing of antiparallel laser beams leads to energy being transferred from the high frequency electromagnetic wave to the lower frequency electromagnetic wave and to the beat frequency electrostatic wave. The electrostatic wave is damped by collisional, and collisionless mechanisms and is eventually converted to plasma thermal energy. The ratio of energy transferred to the electromagnetic wave to energy transferred to the electrostatic wave is given by the Manley-Rowe relations (5.17) as (ω_2/ω_3). This process has often been considered as a possible mechanism for heating underdense plasmas. From the Manley-Rowe relations beat frequency heating can be much more efficient for ω_3 close to the Langmuir frequency than it is for ω_3 close to the ion-acoustic frequency.

In this chapter, some applications of the amplification of the lower frequency electromagnetic wave will be considered. A device is proposed which could be capable of efficiently amplifying coherent radiation in the 11-14μm range. This same device could also be capable of amplifying very short pulses of infrared radiation.

6.1 A Plasma-Laser Amplifier in the 11-14μm Wavelength Range

A laser amplifier in the infrared is of interest for applications in laser heating of plasmas confined magnetically in solenoids and recently for nuclear isotope separation. In this study a laser amplifier is proposed which relies on a 10.6μm laser to serve as a pump for amplifying laser beams in the range 11-14 μm with a high conversion efficiency. The proposed scheme is shown to be ideally suited for amplification of radiation in the range 12-13μm which includes the 12.1μm line which is of interest in nuclear isotope separation (see Jensen et al.[32]).

The basis of operation for the proposed amplifier is the nonlinear interaction in a plasma between a strong 10.6μm laser beam which serves as a pump and a lower frequency laser beam which is to be amplified. The two beams force an electron plasma wave at their beat frequency.

The analysis of Chapter 5 shows that for a plasma with a temperature greater than ∿ 100eV, the mechanism for the transfer of energy to the lower frequency laser beam is very insensitive to variations in the plasma density and temperature. That is, this technique for transferring energy to a lower frequency laser beam does not need to be restricted only to cases where the difference frequency of the laser beams is very close to the Langmuir frequency. Based on this observation, a laser amplifier in the far infrared with plasma serving as a nonlinear mixing medium appears experimentally realizable. The proposed scheme is essentially a parametric amplifier with the crystal replaced by a plasma. However, this amplifier would operate at much higher beam intensities than parametric amplifiers which use a crystal as their beam mixing medium. The amount of energy transferred from the pump to the lower frequency waves is found to obey a Manley-Rowe type of relationship, equation (5.17).

$$\frac{W_1}{\omega_1} = \frac{W_2}{\omega_2} = \frac{W_3}{\omega_3}$$

where W_1 is the energy transferred from the high frequency beam, W_2 and W_3 are the energies transferred into the low frequency beam and the plasma wave respectively. ω_1 and ω_2 are the frequencies of the high and low frequency electromagnetic beams respectively and $\omega_3=\omega_1-\omega_2$ is the frequency of the driven plasma wave.

A schematic of a simple optical amplifier is shown in Figure 6.1(a). In this device plasma is radially confined by a solenoidal magnetic field. Even though the 10.6μm laser beam enters the solenoid at a slight angle it will be guided down the axis of the solenoid due to a density minimum on the axis. Such a density minimum could be created during the plasma formation or by the CO_2 laser itself (see Steinhauer[33], 1976). The required physical length of such a device is dependent on the plasma density and temperature profiles as well as the wavelength and initial intensity of the beam to be amplified. These factors are discussed in detail later.

Certain advantages can be gained by placing the amplifier in an optical resonator. Two possible configurations are shown in Figures 6.1(b) and 6.1(c). The mirror configuration in Figure 6.1(b) is the same as that of a laser. Part of the low frequency beam is reflected back into the plasma by partially reflecting mirror M_2. The low frequency beam propagating antiparallel to the 10.6μm beam is amplified at the expense of the 10.6μm beam. The portion of the low frequency beam propagating parallel to the 10.6μm beam does not interact with the 10.6μm beam (see Capjack and James[30]). An optical resonator enables the low frequency beam (frequency ω_2) to be of relatively high intensity and thus enhances the energy transfer process. A second major advantage of placing the amplifier in an optical resonator is that the low frequency beam can continue to be emitted long after the initial low frequency beam has shut off. Thus a short Q-switched pulse signal can be used to start the process which will then continue long after the initial low frequency signal has died away. There is also a possibility that the low frequency signal could be generated spontaneously, thus eliminating the need for the injected signal. In this case however precise frequency control could not be achieved. The principle of operation of the device with the mirror configuration shown in Figure 6.1(c) is much the same as that of Figure 6.1(b). The main difference is that the low frequency beam now only traverses the plasma in one direction. This has the advantage of reducing the absorption of the beam by the plasma through inverse Bremsstrahlung absorption.

For either of the mirror configurations the ratio of the intensity of the low frequency beam as it enters the plasma to that of the high frequency beam as it enters the plasma is given by:

$$\frac{I_2(L)}{I_1(0)} = \frac{\omega_2}{\omega_1}\frac{R}{(1-R)}$$

where R is the fraction of energy reflected by the partially reflecting mirror, $I_1(x)$ and $I_2(x)$ are the intensities of the pump beam and the beam to be amplified respectively in W/cm^2, and L is the length of the solenoid. In deriving this relation it has been assumed that inverse Bremsstrahlung absorption is negligible and that all of the energy in the high frequency wave has been transferred to the two lower frequency waves.

Optimal efficiency in energy conversion from the high frequency beam can be achieved by making the physical length of the solenoid longer than the absorption length of the first beam. By neglecting absorption through inverse Bremsstrahlung (valid for high plasma temperatures since absorption length as a $T_e^{3/2}$ dependence) a theoretical estimate of the spatial dependence of the intensities of the two beams in the plasma can be written as:

$$\frac{\partial I_1(x)}{\partial x} = - \alpha I_1(x) I_2(x),$$

$$\frac{\partial I_2(x)}{\partial x} = - \frac{\omega_2}{\omega_1} \alpha I_1(x) I_2(x) \tag{6.1}$$

These equations are obtained from (5.40) by assuming $\Gamma_1 = \Gamma_2 = 0$, $c_1 = - c_2 = c$ and $|k_2/k_1| = (\omega_2/\omega_1)$. The last two relations are valid if $\omega_{1,2} >> \omega_{pe}$.

Note that the high frequency beam propagates in the +x direction and is attenuated while the low frequency beam propagates in the -x direction and grows. Solving (6.1) for $I_1(x)$

$$I_1(x) = \frac{\alpha b I_1(0)\exp(-\alpha bx)}{\alpha b+\alpha(\omega_2/\omega_1)I_1(0)[1-\exp(-\alpha bx)]} \qquad (6.2)$$

where $b=[I_2(L)-(\omega_2/\omega_1)I_1(L)]$.

Theoretical estimates of the parameter α have been made in Chapter 5. Figure 5.5 indicates that for a 12 μm beam mixing with a 10.6 μm beam, α could be expected to be greater than 10^{-12} for a density near 10^{17} cm^{-3} and a temperature near 200 eV. Figure 5.6 shows that much higher temperatures and densities are needed to achieve a broad resonance and high value of α for the amplification of a 14 μm line. However high temperatures may be achieved in such a device since the plasma is heated by both inverse Bremsstrahlung absorption of the electromagnetic waves and by the beat frequency mixing process.

If a 14 μm beam is being amplified and $\alpha = 10^{-12}$, equation (6.2) predicts that the first beam will be 90% absorbed in 20 cm if both beams enter the plasma with an intensity of 10^{11} W/cm^2. If the first beam enters the plasma with an intensity of 10^{12} W/cm^2 and the second beam enters with an intensity of 10^9 W/cm^2 the first beam will be 90% absorbed in 60 cm.

This work could be extended through the development of a computer subroutine which numerically solves equations (5.40) in a time and spatially varying plasma and which is then interfaced with the two dimensional simulation described in Chapter 2. In this way the plasma parameters could be calculated self-consistently with the beat frequency heating rate.

6.2 Short Pulse Amplification

In this section, the possibility of using the beat frequency mixing process to amplify and temporally shorten pulses of infrared laser radiation is considered. Consideration is given to cases where the beat frequency is close to the electron plasma frequency and to cases where the beat frequency is close to the ion-acoustic frequency. The basis of operation for such a device is as follows: A signal oscillator injects a pulse of electromagnetic radiation into the plasma antiparallel to the pump beam. The beat frequency mixing process then causes an energy transfer out of the pump beam, and into the signal to be amplified and the electrostatic mixing mode. If this mixing occurs rapidly the pump beam may be rapidly depleted causing only the leading edge of the signal beam to be amplified. This leads to a comparatively long signal pulse growing into a sharp burst of intense radiation. Also considered is the case where this device is used to amplify an ultra-short ($\lesssim$ 100ps) pulse of laser radiation. Ways of generating such short pulses have been developed by Yablonovitch and Goldhar[34], and later by Fisher and Fieldman[35].

In the previous section it was suggested that a plasma could be a useful nonlinear medium for the parametric amplification of infrared radiation. This analysis assumed a steady state condition (no temporal dependence) for the plasma and interacting beams and consequently is not valid for describing the amplification of short pulses of radiation. While the former analysis considered only the case where the beat frequency mode was an electron plasma mode, the present analysis concentrates more on the case where the beat frequency mode is an ion-acoustic mode.

Again, Figure 6.1(a) can serve as a schematic diagram for the proposed amplifier. For purposes of this thesis, it is assumed that a pulse of radiation with a wavelength of approximately 10.6μm is to be amplified. If the pump beam has a wavelength of approximately 9.6μm and the pulse to be amplified has a wavelength of 10.6μm, the beat frequency of these two waves is close to the electron plasma frequency of a plasma of density $10^{17}cm^{-3}$. If the appropriate plasma conditions exist an electron plasma wave can be driven and the 10.6μm pulse is amplified at the expense of the 9.6μm beam. Similarly, if the pump beam has a wavelength of 10.247μm (R(20) transition for a CO_2 laser), and the pulse to be amplified has a wavelength of 10.2605μm (R(18) transition for a CO_2 laser), the beat frequency of these two waves is close to the ion-acoustic frequency of a plasma of 350eV. If the appropriate plasma conditions exist a large ion-acoustic wave can be driven as a beat frequency mixing mode.

The energy that is transferred from the pump beam to the pulse to be amplified is dependent on the device length, the mixing rate, the frequencies of the two beams as well as inverse Bremsstrahlung absorption rates. If inverse Bremsstrahlung absorption is neglected, and it is assumed that the mixing rate is sufficiently rapid so that essentially all of the energy is transferred out of the higher frequency beam, the principle of conservation of energy and the Manley-Rowe relations yield the following expression for the energy gained by the pulse.

$$\Delta E = 2I_1A\frac{\omega_2}{\omega_1}\quad\frac{1}{c}\;L \tag{6.3}$$

where ΔE is the energy gained by the pulse, I_1 is the average cross-sectional intensity of the pump beam, A is the cross-sectional area of the pump beam, c is the speed of light and L is the plasma length. In the design of this device, the device length, the pump beam intensity, and the plasma conditions must be chosen in such a way that Brillouin backscatter is not a serious problem. Since this device is intended to amplify back-travelling radiation, threshold conditions for parametric instabilities must, by definition, be exceeded. Since these instabilities must grow from thermal plasma fluctuations (which may be incoherent) no problem will be created if the device is sufficiently short. Exactly what conditions must be met before these instabilities create a problem cannot be determined at the present time. For the purposes of this analysis, it is assumed that a sufficient condition to prevent problems with parametric instabilities is:

$$\exp(\alpha_s I_1 L) \lesssim 10^4 \tag{6.4}$$

where α_s is the growth rate for the back-scattered radiation. α_s can be found as the maximum on a curve of α versus ω_3 for given plasma conditions. Since this analysis uses the linearized Vlasov equation, we must be careful to ensure that plasma and beam conditions are always such that:

$$\frac{\tilde{n}_e}{n_o} << 1 \tag{6.5}$$

where $\tilde{n}_e$ is the magnitude of the density fluctuations due to the beat frequency mixing process and n_o is the equilibrium density.

Again, certain advantages can be gained by placing the amplifier in an optical resonator such as those illustrated schematically in Figures 1(b) and 1(c). This can greatly increase the interaction length for a pulse allowing it to grow to larger amplitudes. Since higher amplitude pulses tend to sharpen quicker due to pump depletion, optical resonators could lead to shorter pulses. After the signal oscillator has injected one pulse into the amplifier a whole series of pulses would be emitted. The temporal separation of these pulses would be L_p/c where L_p is the optical pathlength of one round trip in the device. In order to prevent parametric instabilities from becoming a problem in this case, a saturable absorber would have to be placed in the path of the pulse.

6.2.1 Quasi-Static Analysis

From Section 5.3, equations (5.40) are valid if the resonance width of $f(k,\omega)$ is much narrower than the resonance width of $D(k,\omega)$ so that equation (5.34) can be approximated. The resonance width of $D(k,\omega)$ is of the order of the damping rate of the electrostatic mode γ_L, and the resonance width of $f(k,\omega)$ is $\Delta\omega_3$. Therefore, equations (5.4) are valid for:

$$\Delta\omega_3 << \gamma_L$$

$\Delta\omega_3$ can be approximated as:

$$\Delta\omega_3 = \Delta\omega_1 + \Delta\omega_2$$

where $\Delta\omega_1$ is the frequency spread of the pump beam and $\Delta\omega_2$ is the frequency spread of the pulse being amplified. Considering the pump beam and the pulse to have minimum frequency spreads, $\Delta\omega_1 << \Delta\omega_2$ so:

$$\Delta\omega_3 \simeq \Delta\omega_2$$

Thus the quasi-static analysis is valid if:

$$\Delta\omega_2 << \gamma_L$$

or

$$\tau_p >> \gamma_L^{-1} \qquad (6.6)$$

where τ_p is the temporal width of the pulse. When (6.6) is true, the evolution of the pulse can be found by solving (5.40):

$$[\partial_t + c_1 \partial_x + 2\Gamma_1] I_1 = \alpha c_1 I_1 I_2$$
$$[\partial_t + c_2 \partial_x + 2\Gamma_2] I_2 = - \left| \frac{k_2}{k_1} \right| \alpha c_1 I_1 I_2 \tag{6.7}$$

In this analysis it is assumed that the plasma is well underdense so that:

$$c_1 = - c_2 = c = 3x10^{10} cm/sec$$
$$\left| \frac{k_2}{k_1} \right| = \frac{\omega_2}{\omega_1} \tag{6.8}$$

Equations (6.7) can be integrated numerically, however it is useful to notice that the amplification factor of the back travelling pulse in the pulse front (before pump depletion becomes important) is simply given by $\exp[\alpha I_1 \ell]$ where ℓ is the distance this portion of the pulse has travelled in the plasma.

Figure 5.4 is a plot of α as a function of n_o for various plasma temperatures for the case of λ_1=9.6μm and λ_2=10.6μm. A value of $\alpha > 4x10^{-12}$ is expected for the case of T_e close to 100eV and plasma density between $7x10^{16}$ and $8x10^{16} cm^{-3}$. Chapter 3 discusses a M.H.D. computer simulation of a laser heated plasma in a one meter solenoid with an intial density of $7.8x10^{16} cm^{-3}$. After 0.5μ-sec of heating with a 1GW laser the plasma had an on axis density of $7.5x10^{16} cm^{-3}$ and an electron temperature between 95 and 100eV over a distance of more than

80cm. In other words, the plasma conditions are such that almost optimum values of α exist over most of the length of the one meter solenoid. Figure 5.3 shows α as a function of ω_3 for these plasma conditions. It is seen that the peak value of α due to the Brillouin process is less than the value of α predicted for the Raman process. Consequently, Brillioun scattering would not be expected to be a problem for this case.

Another M.H.D. simulation reported in Chapter 3 is that of a laser heated plasma in a one meter solenoid and an initial density of $3.0\times10^{17}cm^{-3}$. After one μsec of heating with a one GW laser the plasma has a density between 2.4×10^{17} and $2.5\times10^{17}cm^{-3}$, an electron temperature of between 120 and 180eV, and an ion temperature of between 100 and 140eV over an 80cm length of the one meter solenoid. Figure 5.2 shows α as a function of ω_3 for these plasma conditions. It is seen that a large value of α is achieved due to mixing with the ion-acoustic mode if ω_3 is close to $2.4\times10^{11}sec^{-1}$. Beat frequency mixing of CO_2 laser beams with wavelengths of λ_1 = 10.2470 (R(20)) and λ_2 = 10.2605, R(18)) yields $\omega_3 = 2.42\times10^{11}sec^{-1}$. From Figures 5.2, 5.7, 5.8 and 5.9 it is seen that the resonance for this mode is very broad and a value of $\alpha > 4\times10^{-12}$ could be expected over at least 80cm of the 1 meter solenoid.

The above discussion indicates that suitable plasma conditions could be attained in a laboratory to yield values of $\alpha > 4\times10^{-12}$ for the mixing of radiation with wavelengths of either 9.6μm and 10.6μm, or 10.247μm and 10.2605μm. For either case the evolution of the interacting waves should be similar, as predicted by equations (6.7). A condition for the validity of the quasi-static approximation is given by equation (6.6). The plasma conditions assumed for the beat frequency mixing of 9.6μm and 10.6μm radiation (T_e=100eV and n_o=7.75×10^{16} cm^{-3}) imply a value of $k_3\lambda_D$=0.33 and γ_L=3.8×10^{11} sec^{-1}. For the case of mixing radiation with wavelengths of 10.247μm a quasi mode (damping rate is almost as large as the frequency) is driven and $\gamma_L \approx 3.3\times10^{11}$ sec^{-1}. Thus for either of these cases the quasi-static approximation is valid if the temporal width of the pulse to be amplified, τ_p, satisfies

$$t_p >> 3\times10^{-12} \text{ sec} \tag{6.9}$$

In order to illustrate the amplification of a pulse using this process, equations (6.7) have been integrated numerically assuming the following conditions:

λ_1 = 9.6μm

λ_2 = 10.6μm

n_o = 7.5×10^{16} cm^{-3}

Plasma length = 80cm

Initial pump intensity = 1.6×10^{10} W/cm^2

Input pulse intensity = $I_2(t)$ where

$$I_2(t) = \begin{cases} 2\times10^9\ I_{2p}\cdot t & t < .5\times10^{-9} \text{ sec} \\ 2\times10^9\ I_{2p}(10^{-9}-t) & .5\times10^{-9} < t < 10^{-9} \\ 0 & t > 10^{-9} \text{ sec} \end{cases}$$

That is the injected pulse is of triangular form rising to a peak power of I_{2p} W/cm^2 at 0.5 ns and falling off to zero again at 1 ns. The above parameters yield α=$5.7x10^{-12}$ and in the absence of pump depletion the whole pulse would be amplified by a factor of 1500.

The results of these simulations are illustrated graphically in Figures 6.2, 6.3, and 6.4. In each of these figures the intensity is plotted on a logarithmic scale as a function of axial distance down the solenoid axis at three different times. Plasma is assumed to exist only in the interval 0-80cm and α is therefore set to zero in the interval of 80-120cm (i.e. no mixing takes place in this region). Figure 6.2 illustrates the pulse propagating down the solenoid for a case where there was no beat frequency mixing. The pump beam intensity was set to zero for this case. Figure 6.3 illustrates the growth of a pulse which initially had a peak intensity of 10^8 W/cm^2. In this case the peak pulse power has been amplified by a factor of about 250. This is considerably less than what would be attained if pump depletion were not important. The pump intensity at the three different times is also plotted in this figure. Figure 6.4 is a similar plot except the peak power in the pulse was 10^9 W/cm^2 before it was injected into the amplifier. The interesting feature of this calculation is that the front end of the pulse has sharpened considerably during amplification. Initially the pulse rose from zero to its peak power in 0.5 ns. After amplification it rises to peak power in 0.13 ns. In fact the rise time of this pulse is becoming sufficiently rapid that the validity of the quasi-static approximation is becoming questionable. The reason the pulse sharpens in this method of amplification is that the pump is depleted near the leading edge of the pulse. As a result only the leading edge of the pulse is amplified. The higher the intensity of the injected pulse the steeper the leading edge becomes during amplification. For pulses with very sharp leading edges, the quasi-static approximation breaks down and a more sophisticated transient analysis must be done.

6.2.2 Transient Analysis

In Chapter 5 a set of equations which describe the evolution of a pulse of electromagnetic radiation propagating anti-parallel to another electromagnetic beam with a different frequency in a plasma were derived. These equations ((5.65), (5.66), and (5.70)) account for transient effects in the electrostatic mode. Before these equations are solved two additional approximations are made.

1. In the integration of (5.70)

   $A_1(x+c_2\tau, t-\tau) \approx A_1(x+c_2\tau, t)$. This is valid when A_1 and $\tilde{n}_e$ are slowly varying functions in the moving coordinate system.

2. $c_2 = -c_1$. This is valid if $\omega_2 \gg \omega_{pe}$.

With these approximations, the equations to be solved are summarized below. Primes have been dropped for clarity.

$$[\partial_t + 2c_1\partial_x + \Gamma_1]\, A_1(x,t) = R_1 \tilde{n}_e(x,t)\, A_2(x,t) \tag{6.10}$$

$$[\partial_t + \Gamma_2]\, A_2(x,t) = R_2\, \tilde{n}_e^*(x,t)\, A_1(x,t) \tag{6.11}$$

$$\tilde{n}_e(x,t) = \frac{1}{\sqrt{2\pi}} \int_0^{x/c_1} A_1(x-c_1\tau, t)\, A_2(x-c_1\tau, t)\, e^{i\omega_3\tau} D(k_3,\tau)\, d\tau \tag{6.12}$$

where

$$R_1 = -\left(\frac{i}{2}\right)\frac{\omega_{pe}^2}{\omega_1 n_o} \qquad R_2 = -\left(\frac{i}{2}\right)\frac{\omega_{pe}^2}{\omega_2 n_o} \tag{6.13}$$

$$\Gamma_1 = \frac{\nu_{ei}\omega_{pe}^2}{2\omega_1^2} \qquad \Gamma_2 = \frac{\nu_{ei}\omega_{pe}^2}{2\omega_2^2} \tag{6.14}$$

$$\nu_{ei} = 2.9\times10^{-6}\, n_o \,\ln\Lambda / T_e(eV)^{3/2} \tag{6.15}$$

$$c_1 = \frac{c^2 k_1}{\omega_1} \tag{6.16}$$

$$D(k_3,\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} D(k_3,\omega)\, e^{-i\omega\tau}\, d\omega \tag{6.17}$$

$$D(k_3,\omega) = \frac{k_3^2}{4\pi m_e c^2}\;\frac{\chi_e(1+b\chi_i)}{1+b\chi_e+b\chi_i} \tag{6.18}$$

$$b = 1 - a \qquad a = \frac{i\omega\nu_{ei}}{\omega_{pe}^2} \tag{6.19}$$

and χ_e and χ_i are defined in Appendix B.

The above equations are equally valid for beat frequency mixing with a difference frequency near the electron plasma frequency or the ion-acoustic frequency. However, in the numerical integration a timestep is limited to being much less than the period of the beat wave and consequently a prohibitive amount of computer time would be required to integrate the equations if the beat frequency is close to the electron plasma frequency. In this section solutions for the case of the beat frequency being close to the ion-acoustic frequency are reported.

For the purposes of these computations the assumed plasma conditions correspond to those of the MHD simulation reported in Chapter 3 with an initial density of 3.0x10[illegible]. Since Figures 5.8 and 5.9 indicate that the ion mode is very broad, it is reasonable to approximate the plasma being uniform. A plasma with T_e=160eV, T_i=120eV, and n_o=2.5x10^{17}cm^{-3} is assumed. The wavelength of the pump beam is 10.247μm (λ_1) and the wavelength of the pulse to be amplified is set to 10.2605 μm (λ_2). This yields a difference frequency of ω_3=2.42x10^{11}sec^{-1} and closely corresponds to a maximum in the plot of α versus ω_3 in Figure 5.2. For these conditions α=6.06x10^{-12}. By assuming an interaction length of 80 cm, the input pump intensity must be less than 1.9x10^{10} W/cm^2 to satisfy equation (6.4).

Figure 6.5 shows the real and imaginary parts of $D(k_3,\omega)$ as a function of ω for T_e=160eV, T_i=120eV and n_o=2.5x10^{17}cm^{-3}. Figure 6.6 shows $D(k_3,\tau)$ as a function of τ for these same conditions. It is noted that the real part of $D(k_3,\omega)$ is even, while the imaginary part is odd as is required to make $D(k_3,\tau)$ real. The plot of $D(k_3,\tau)$ versus τ shows that the density fluctuation level at time t is only dependent on the driving forces in the time interval from t-T to t where T is the period of the beat frequency wave. This is characteristic of a heavily damped oscillator. To illustrate the effects of reducing the damping rate similar plots of $D(k_3,\omega)$ and $D(k_3,\tau)$ are shown in Figures 6.7 and 6.8 for the case of T_e=160eV, T_i=20eV, and n_o=2.5x10^{17}. Here a much sharper structure is seen in the frequency spectrum and $D(k_3,\tau)$ is seen to damp out after several periods.

To illustrate the importance of transient effects without the complications of pump depletion and inverse Bremsstrahlung absorption, a calculation has been done which assumes an input pulse which rises instantaneously to a constant value of 10^5 W/cm^2 was injected into the solenoid antiparallel to a beam with an intensity of 1.5×10^{10} W/cm^2. For this calculation the effects of inverse Bremsstrahlung absorption have been neglected. The initial pulse intensity (10^5 W/cm^2) is sufficiently low that the effects of pump depletion are negligible. Figures 6.9 and 6.10 show I_1, I_2, and $|\tilde{n}_e|$ as a function of $\tilde{x}$ (where $\tilde{x}$ is the distance in cm from the pulse front) after the pulse front has travelled 40cm and 80cm in the plasma. For the case of Figure 6.10 a quasi-static analysis predicts that the pulse front should have experienced an amplification of a factor of 1440 and that the portion of the pulse near $\tilde{x}=3$ cm, which has only travelled 77 cm in the plasma, should be amplified by a factor of 1090. The transient analysis and quasi-static analysis agree at $\tilde{x}=3$ cm but transient effects lead to a smaller amplification in the pulse front. Comparison of Figures 6.9 and 6.10 shows that the further the pulse has travelled in the plasma the larger the value $\tilde{x}$ must be before the transient analysis agrees with the quasi-static analysis.

A similar calculation has been made to illustrate the importance of pump depletion. Again, it is assumed that the injected pulse rises to a constant value instantaneously, however this time the initial intensity of the injected beam is 10^{10} W/cm^2 and the intensity of the pump is 1.8×10^{10} W/cm^2. For these conditions a quasi-static analysis of the pulse front predicts an amplification of a factor of 6163 after it has interacted over a region of 80cm. Figures 6.11 to 6.14 show I_1, I_2 and $2|\tilde{n}_e|$ has a function of $\tilde{x}$ after the pulse has travelled 40, 60, 70 and 80cm respectively. From Figure 6.14 it is seen that the maximum amplification is a factor of 100 and occurs at $\tilde{x}=0.6$cm. It is interesting to note that up to the time the pulse has travelled 70cm in the plasma $I_1(x)$ is a monotonically decreasing function, whereas after the pulse has travelled 80cm $I_1(\tilde{x})$ experiences some regrowth behind the peak of the pulse. If the pulse is sufficiently sharp the phase relationship between the electrostatic wave and the two electromagnetic waves can be such that energy is transferred from the lower frequency waves to the high frequency wave in this region.

A calculation with an injected pulse with a more realistic initial profile has also been made. The injected pulse is assumed to be of triangular shape rising from 0 at t=0 to 10^{10} W/cm^2 at t=50ps and falling off to zero at 100 ps. The intensity of the pump wave is again assumed to be 1.8×10^{10} W/cm^2. Figures 6.15 and 6.16 show I_1, I_2 and $|\tilde{n}_e|$ as a function of $\tilde{x}$ after the pulse has travelled 40 and 80 cm in the plasma. It can be seen that the maximum value of the signal intensity has moved ahead and has a value of about 80 times the peak value before amplification. It is also seen that the pulse width has decreased from 50 ps to about 30 ps during amplification.

The above analysis indicates that the process of beat frequency mixing in a plasma could be used for the amplification, and in some cases, the shortening of pulses of electromagnetic radiation. This type of analysis could also be used to examine Brillioun backscattering from a laser heated plasma.

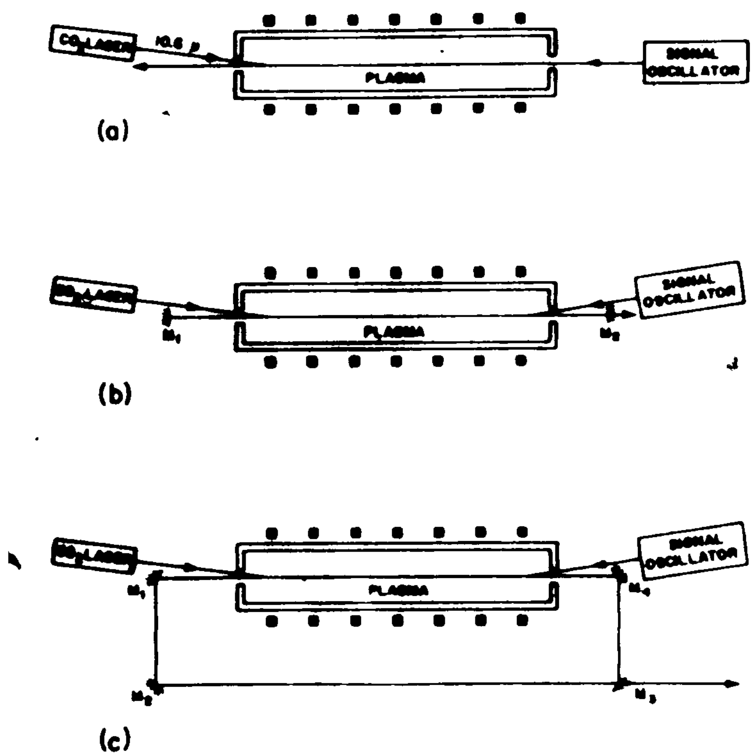

Figure 6.1 Schematic of optical amplifier. Mirrors in (b) and (c) make an optical resonator and increase the intensity of the low frequency beam.

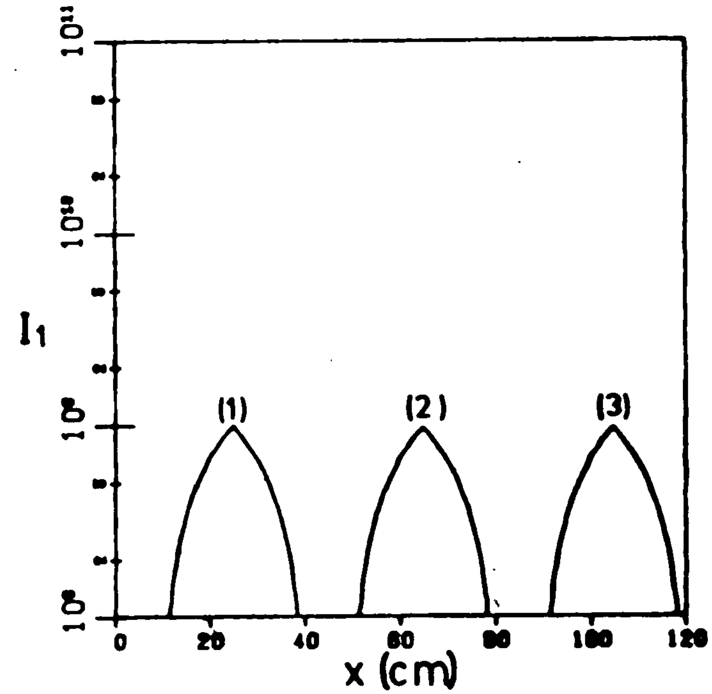

Figure 6.2 I_2 in W/cm^2 as a function of axial position, x, at three different times. (1) t=1.33ns, (2) t=2.67ns; and (3) t=4ns. No beat frequency mixing in this case.

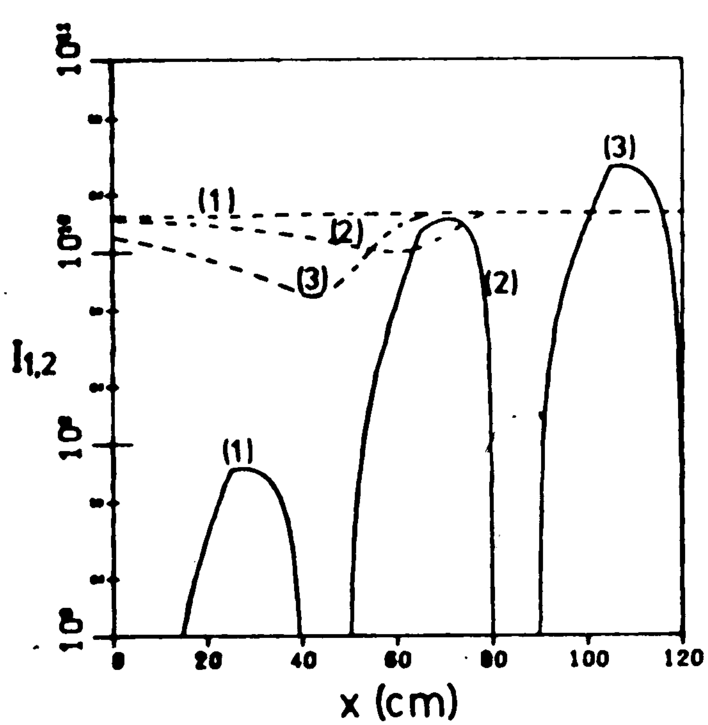

Figure 6.3 I_1 (dashed curves) and I_2 (solid curves) in W/cm^2 as a function of axial position, x, at three different times. (1) t=1.33ns, (2) t=2.67ns, and (3) t=4ns. Initial peak pulse intensity is 10^8 W/cm^2.

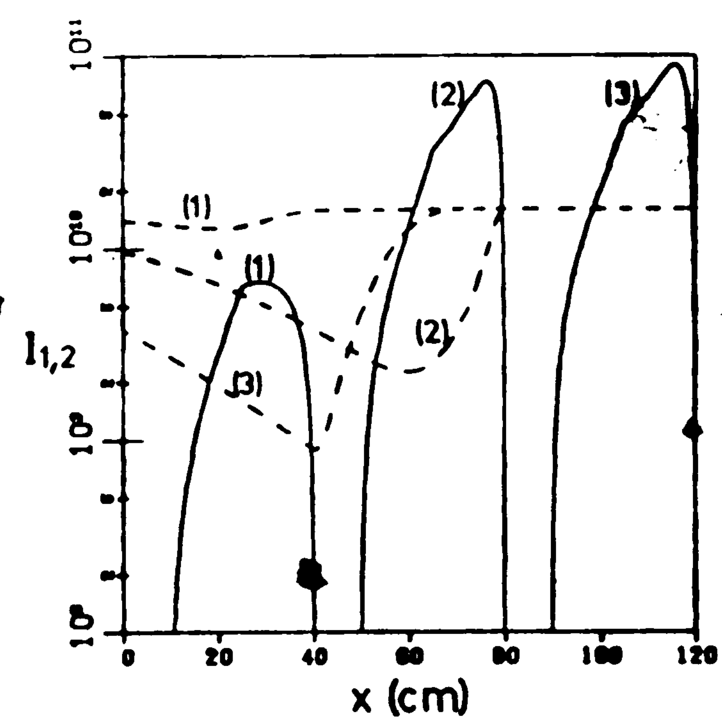

Figure 6.4 I_1 (dashed curves) and I_2 (solid curves) in W/cm^2 as a function of axial position, x, at three different times. (1) t=1.33ns, (2) t=2.67ns, (3) t=4ns. Initial peak pulse intensity is 10^9 W/cm^2.

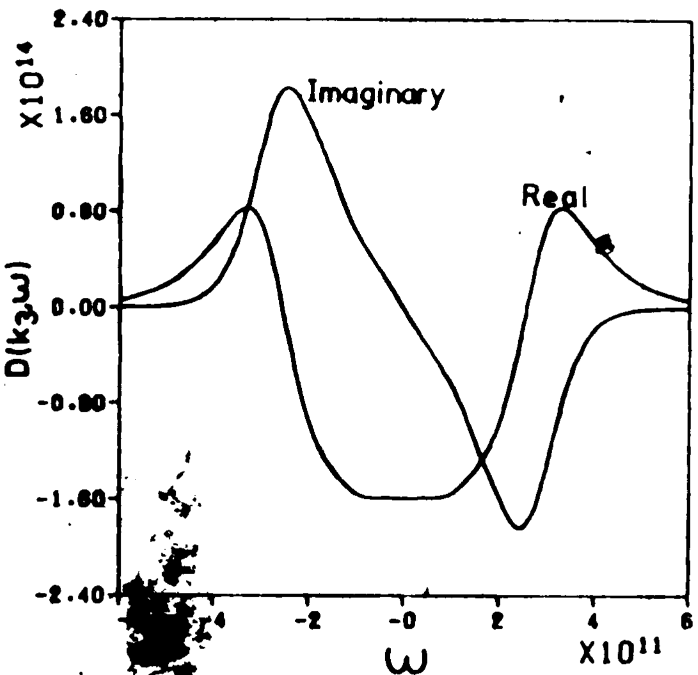

Figure 6.5 Real and imaginary parts of $D(k_3,\omega)$ vs. ω for $T_e=160eV$, $T_i=120eV$ and $n_o=2.5\times10^{17}cm^{-3}$.

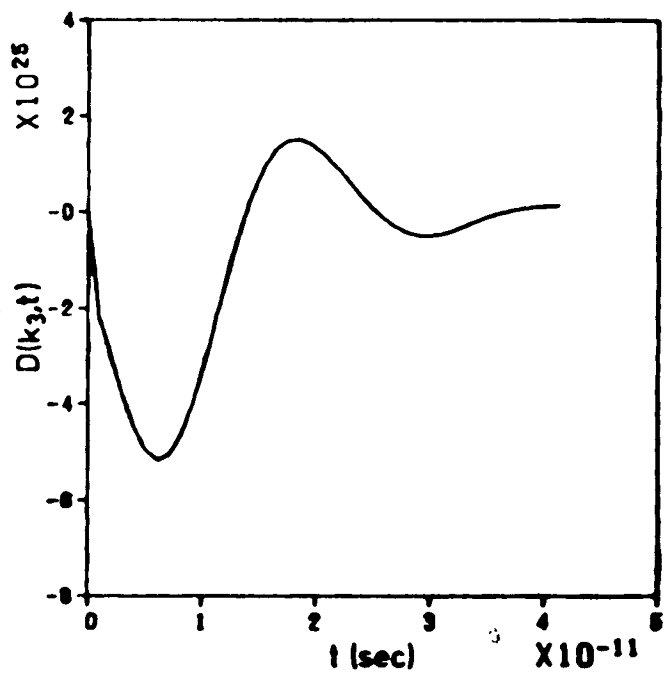

Figure 6.6 $D(k_3,t)$ as a function of t corresponding to $D(k_3,\omega)$ in Figure 6.5.

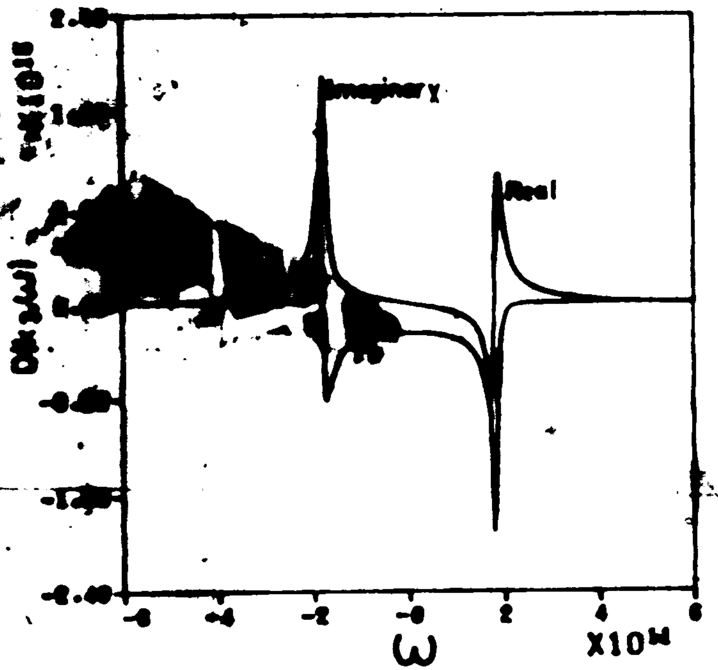

Figure 6.7 Real and imaginary parts of $D(k_3,\omega)$ vs. ω for T_o=160eV, T_i=20eV and n_o=2.5x10^{17}cm^{-3}.

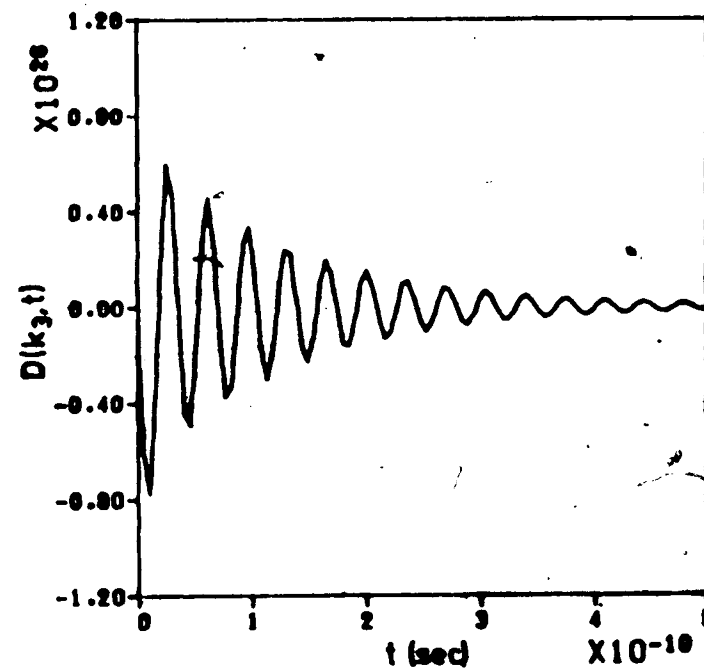

Figure 6.8 $D(k_3,t)$ as a function of t corresponding to $D(k_3,\omega)$ in Figure 6.7.

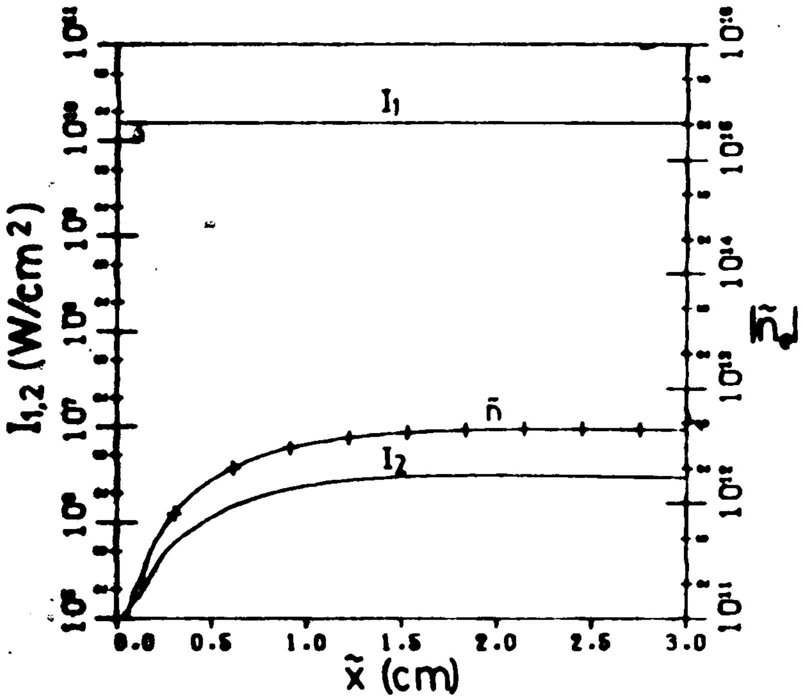

Figure 6.9 I_1, I_2 and $|\tilde{n}_e|$ in cm^{-3} as a function of distance from pulse front, x, after pulse front has travelled 40cm. Initially I_1=1.5x1010 W/cm^2 and I_2=10^5 W/cm^2.

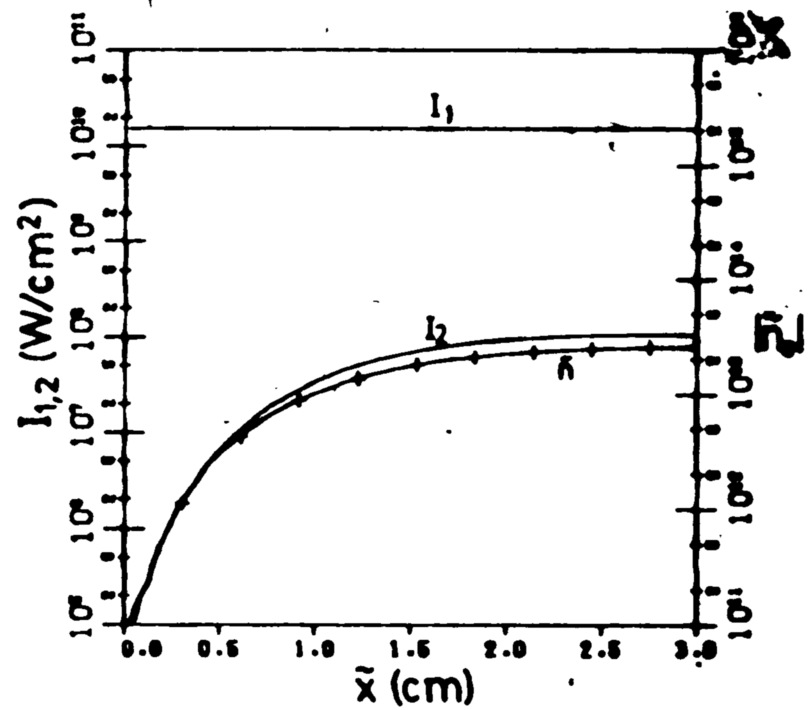

Figure 6.10 Same as figure 6.9 except pulse front has now travelled 80cm.

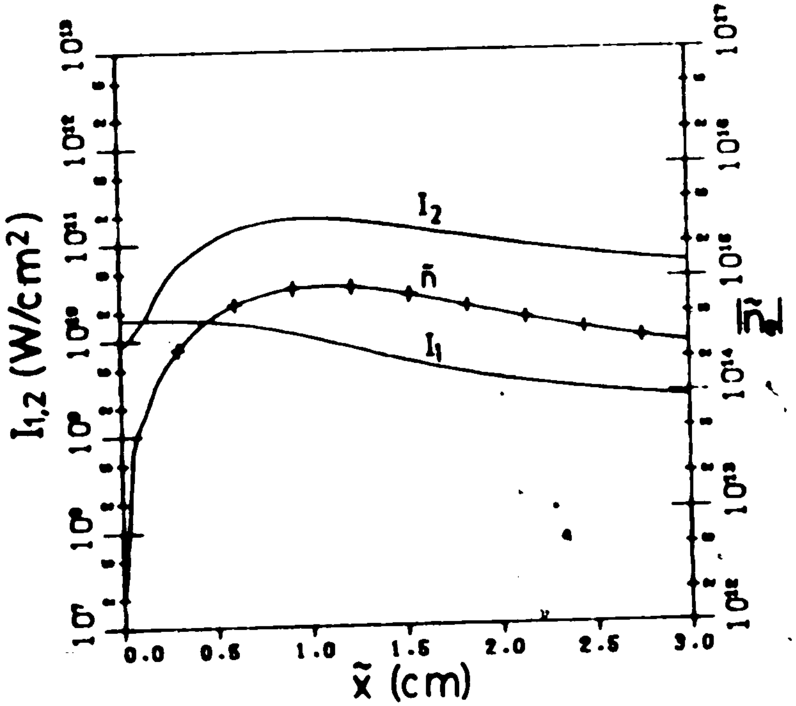

Figure 6.11 I_1, I_2 and $|\tilde{n}_e|$ in cm^{-3} as a function of distance from pulse front, $\tilde{x}$, after pulse front has travelled 40cm. Initially $I_1=1.8\times10^{10}$ W/cm^2 and $I_2=10^{10}$ W/cm^2.

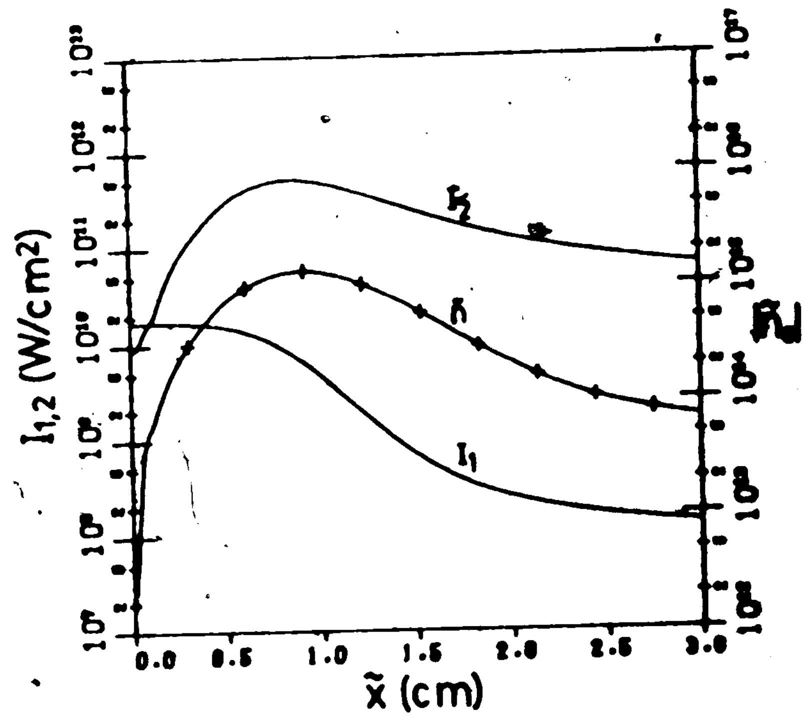

Figure 6.12 Same as figure 6.11 except pulse front has now travelled 60cm.

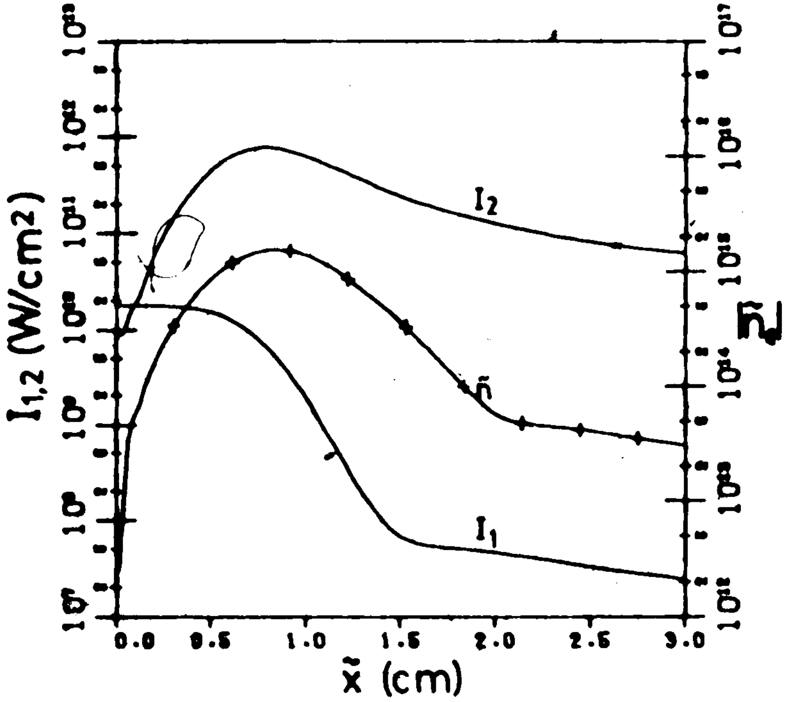

Figure 6.13 Same as figure 6.11 except pulse front has now travelled 70cm.

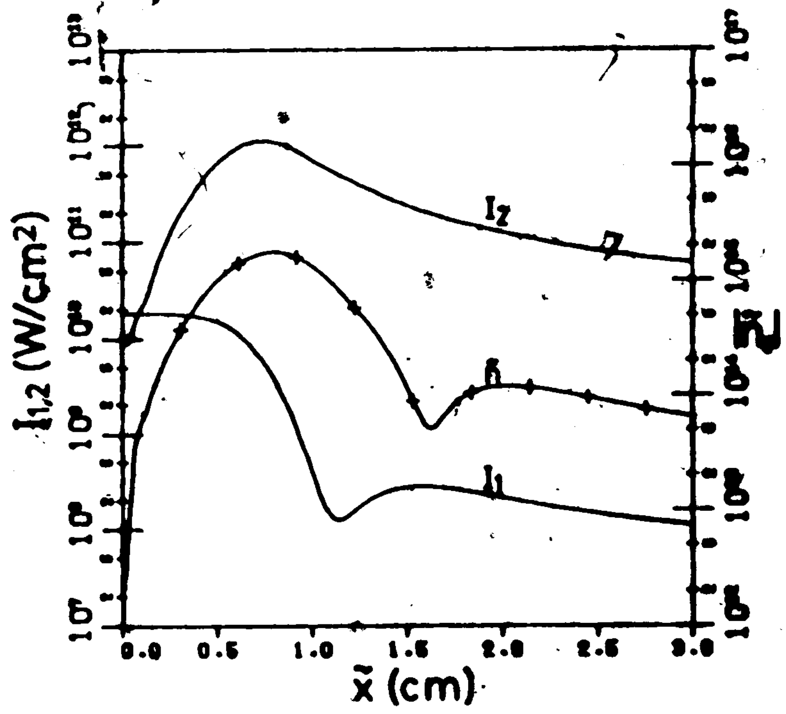

Figure 6.14 Same as figure 6.11 except pulse front has now travelled 80cm.

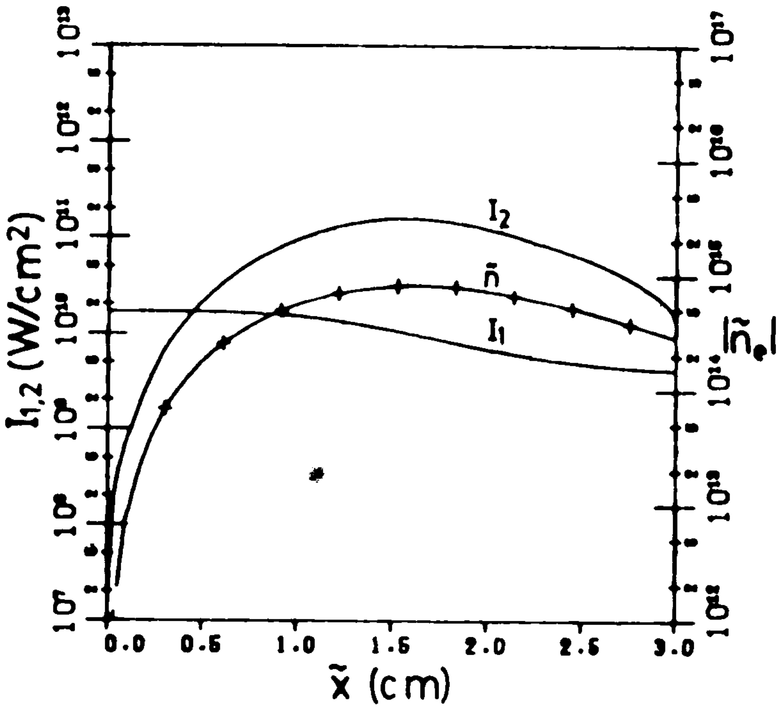

Figure 6.15 I_1, I_2 and $|\tilde{n}_e|$ in cm^{-3} as a function of distance from pulse front, x, after pulse front has travelled 40cm. Initially I_2 is of triangular shape with peak intensity of 10^{10} W/cm^2 at $\tilde{x}$=1.5cm, and I_2=1.8x10^{10}.

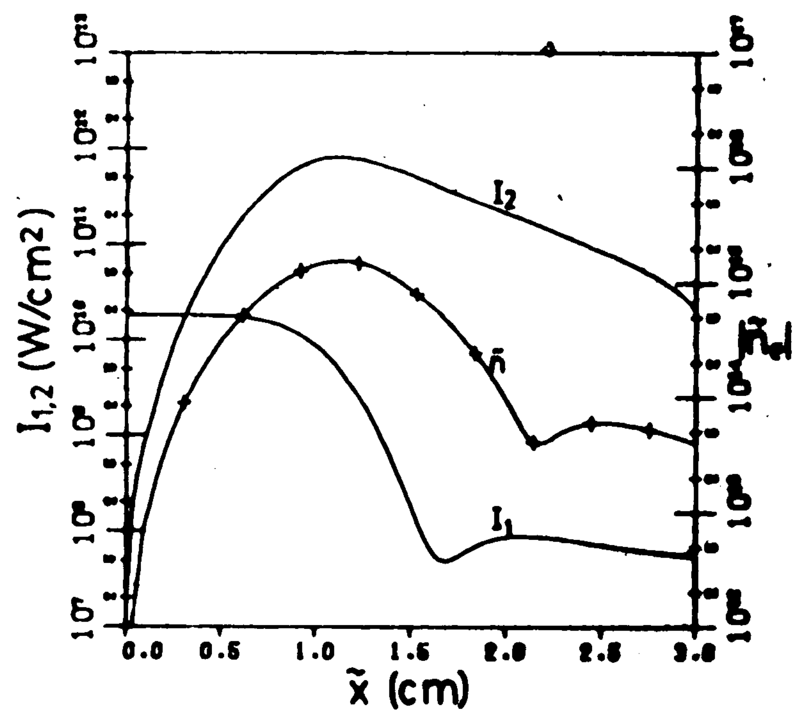

Figure 6.16 Same as figure 6.15 except pulse front has now travelled 80cm.

CHAPTER 7

Conclusions

A two dimensional numerical model of a laser heated plasma in a solenoidal magnetic field has been successfully developed. The model is based on one fluid two temperature MHD equations. By solving the MHD equations in a moving coordinate system defined by the magnetic field lines, numerical diffusion arising from the strong anisotropy of the plasma has been eliminated. This allows for a more coarse computational grid to be used which leads to savings in CPU time.

Care has been taken to make the computer routine as versatile as possible. One example of this is that, to a large extent, initial conditions as well as parameters that control some of the internal processing are specified through an input file. In this way the routine can be a flexible and useful tool for users not familiar with the numerical techniques used to solve the equations.

Results from sample calculations made by this routine indicate that the assumption of radial pressure balance is valid for a wide variety of problems. The routine has been constructed so that full radial dynamics can be calculated or radial pressure balance can be assumed.

The routine has been used to provide numerical backup for gas target experiments performed by A.A. Offenberger et al.[19-21] Results from this study indicate that laser heated gas targets can provide an attractive plasma source for the study of laser-plasma interactions. Current near future plans for this routine include more gas target studies and the numerical modelling of the short (1/2 m) solenoid experiments planned by A.A. Offenberger *et al*.[9].

A plasma parametric amplifier has been proposed for amplifying laser beams in the 11-14 μm wavelength range to very high intensities. The plasma section of the amplifying device can be relatively short (less than one meter) and only a very coarse control on plasma density and temperature needs to be maintained. Because the idler wave is an electron plasma wave and is Landau damped the process of energy transfer from the pump beam to the signal beam becomes irreversible. Further analysis suggests that this device could also be used to amplify and temporally shorten ultra-short pulses of infrared laser radiation. Pulse amplification could take place through beat frequency mixing with a beat frequency near the electron plasma frequency or the ion-acoustic frequency.

BIBLIOGRAPHY

1. J.M. Dawson, A. Hertzberg, R.E. Kidder, G.C. Vlases, H.G. Ahlstrom, and L.C. Steinhauer; Proceedings of the Fourth Conference on Plasma Physics and Controlled Nuclear Fusion Research, Vol. 1 (IAEA, Vienna, 1971).

2. A.L. Hoffman, and G.C. Vlases; Appl. Phys. Lett., 12, 690, (1973).

3. L.C. Steinhauer and P.H. Rose, in Proceedings of the First Topical Meeting on Technology and Controlled Thermonuclear Reactions, edited by G.R. Hopkins, United States Atomic Energy Commission Report CONF-740402 (1974), Vol. 2, p. 63.

4. R.D. Richtmyer and K.W. Morton, Difference Methods for Initial Value Problems, Interscience Publishers (1967).

5. F. Hertweck and W. Schneider, Institut fur Plasmaphysik Report IPP6/90 (1970).

6. W. Schneider, Institut fur Plasmaphysik Report IPP1/24 (1972).

7. H.A.B. Bodin, J. McCartan, I.K. Pasco, W.H. Schneider, Phys. Fluids 15, 1341 (1972).

8. L.C. Steinhauer, Phys. Fluids 19, 738 (1976).

9. A.A. Offenberger - Private Communication.

10. R.D. Milroy, C.E. Capjack, and C.R. James, Plasma Physics 19, 989 (1977).

11. W.F. Krupke, Nineteenth Annual Meeting of the Division of Plasma Physics of the American Physical Society, 6K3.

12. S.I. Braginskii, "Transport Processes in a Plasma", in M.A. Leontovich, Ed., Reviews of Plasma Physics, Vol. 1, Consultants Bureau Enterprises (1965).

13. D.L. Book, "Revised NRL Formulary", Naval Research Laboratories.

14. K. Hain, G. Hain, K.V. Roberts, S.J. Roberts, and W. Koppendorfer, Z. Naturforsch. A 15, 1039 (1960).

15. T.W. Johnston and J.M. Dawson, Phys. Fluids, 16, 722 (1973).

16. D.W. Scudder, Z.A. Pietrzyk, G.C. Vlases, and A.H. Makomaski, 10'th International Shock Tubes and Waves Symposium, Seattle, 1977.

17. J.A. Wesson, in Plasma Physics and Controlled Nuclear Fusion Research, (International Atomic Energy Agency, Vienna, 1966) Vol. 1, p. 233.

18. J.N. McMullin and C.E. Capjack, Phys. Fluids, 20, 1566 (1977).

19. A.A. Offenberger, A. Ng, and M.R. Cervenan, Can. J. Phys. 56, 381 (1978).

20. A.A. Offenberger, A. Ng, and L. Pitt, Phys. Rev. Lett. 40, 873 (1978).

21. A.A. Offenberger, A. Ng, L. Pitt, and M.R. Cervenan, (In Press - Phys. Rev. A (1978)).

22. J.J. Schuss, T.K. Chu, and L.C. Johnston, Princeton Report, PPPL-1394.

23. A. Ng, M.R. Cervenan, and A.A. Offenberger, Optics Communications, 25, 116 (1978).

24. F.F. Chen, Introduction to Plasma Physics, Plenum Press, Chapter 8 (1974).

25. G. Schmidt, Phys. Fluids, 16, 1676 (1973).

26. V. Fuchs, C.R. Neufeld, J. Teichmann, and A.G. Engelhardt, Phys. Rev. Lett., 31, 1110 (1973).

27. A.N. Kaufman and B.I. Cohen, Phys. Rev. Lett. 30, 1306 (1973).

28. B.I. Cohen, Ph.D. Thesis, University of California, Berkeley (1975).

29. B.I. Cohen, M.A. Mostrom, D.R. Nicholson, A.N. Kaufman, C.E. Max, and A.B. Langdon, Phys. Fluids, 18, 470 (1975).

30. C.E. Capjack, and C.R. James, Can. J. Phys. 53, 2606 (1975).

31. T.H. [illegible] Theory of Plasma Waves, McGraw Hill (1962).

32. R.J. Jen[illegible] J.G. Marinuzzi, [illegible]inson, and S.D. Rockwood, Laser Focus, May 1976.

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

34. E. Yablonovitch and J. Goldhar, Appl. Phys. Lett. 25, 580 (1974).

35. R.A. Fisher and B.J. Feldman, Optics Letters, 1, 161 (1977)

36. I. Lindemuth and J. Killeen, J. Comp. Phys. 13, 181 (1973).

37. N.H. Burnett and A.A. Offenberger, J. Appl. Phys. 45, 2155 (1974).

Appendix A Some Programming Details for 2-D MHD Code

This Appendix contains some of the programming details of numerical model described in Chapter 2. All equations are written in dimensionless form as described in Chapter 2. The derived quantities are:

- N - Plasma density, n_o
- TE - Electron temperature, T_e
- TI - Ion temperature, T_i
- V - Axial fluid velocity, v_z
- A - Area between shells
- R - Shell radii
- W - Square of shell radii, R^2
- B - Magnetic field, B_z
- U - Radial fluid velocity times radius, Rv

Any two of A, R, and W can be derived from the third however all three are kept as internal variables in the numerical solution for convenience. When the solution employs full radial dynamics, W is solved for, and A and R are derived from W. When radial pressure balance is assumed A is solved for, and R and W are derived from A. The magnetic field B can also be derived directly from A. Initial conditions specify the magnetic flux in each shell, ϕ_s, and this is a constant. B can then be derived from

$$B = \frac{\phi_s}{A}$$

The difference equations are solved on a computational grid consisting of M shells and NX x-points.

A·1 Grid Structure and Two-Step Scheme

The position at which the variables are calculated on the x, s grid is illustrated in Figure A.1. Note, x and s have been declared as integers in the routine and are used to specify the location of a quantity on the computational grid. The quantities N, V, TE, and TI are calculated at the cell centers while U, W, and R are calculated on the shell boundaries. The quantity A denotes the cross sectional area of a cell at a particular x-point. Quantities that are calculated at the shell centers and are subscripted with an (s,x) are assumed to be at the x'th x-point and between the (s-1)'th and s'th shell boundaries as illustrated in Figure A.1.

All variables are calculated at the same time t. In order to obtain second order accuracy in the timestep size, Δt, a two-step method was used to solve the equations. First the solution at time t is used to find temporary values at time $t + \Delta t/2$. These temporary values are used to calculate transport coefficients and spatial derivatives in the equations. The solution is then advanced a full timestep from t to $t + \Delta t$.

At time t the value of all quantities $\mathcal{Q}$ are stored in arrays denoted by $\mathcal{Q}_{s,x}$ and $(\mathcal{Q}2)_{s,x}$. After the first step in the two step scheme $\mathcal{Q}_{s,x}$ contains the temporary solution at time $t + \Delta t/2$. Based on these values of $\mathcal{Q}_{s,x}$, transport coefficients and spatial derivatives are calculated. Based on these, and the values in $(\mathcal{Q}2)_{s,x}$, the solution is advanced from time t to $t + \Delta t$ and placed in array $\mathcal{Q}_{s,x}$. $(\mathcal{Q}2)_{s,x}$ is then set equal to $\mathcal{Q}_{s,x}$ and the process is repeated. A flow-chart illustrating the gross features of the two-step scheme can be found in Figure A.2.

A.2 Difference Equations

The difference equations used in the routine are based on the partial differential equations in Section 2.4.5. All of the equations are of the form:

$$\dot{Q} = \text{RHS} \tag{A.1}$$

where RHS has terms involving first and second order spatial derivatives and time derivatives of other variables. In the code the equations are solved in a sequence such that the required time derivatives of other variables are evaluated first. Expressing the time derivative of the above equation:

$$\dot{Q} = \frac{Q'_{s,x} - Q^{\circ}_{s,x}}{\delta t} = \text{RHS}$$

or

$$Q'_{s,x} = Q^{\circ}_{s,x} + \delta t \cdot \text{RHS}$$

or (A.2)

$$Q'_{s,x} = Q^{\circ}_{s,x} + \delta t \cdot \dot{Q}_{s,x}$$

where $Q^{\circ}_{s,x}$ is the value of $Q_{s,x}$ at time t and $Q'_{s,x}$ is the value of $Q_{s,x}$ at time $t + \delta t$ and δt is the timestep size. In the two-step scheme $\delta t = \frac{1}{2} \Delta t$ for a half step and $\delta t = \Delta t$ for a full step where Δt is the timestep size for a full step.

Central differences are employed throughout when evaluating spatial derivatives. In all cases the partial derivative of Q with respect to x is written as:

$$\left(\frac{\partial Q}{\partial x} \right)_{s,x} = \frac{Q_{s,x+1} - Q_{s,x-1}}{2\Delta x} \qquad (A.3)$$

where Δx is the grid spacing in the x-direction. In the present version the computational grid is uniform in the axial direction so Δx is a constant. Differencing in the radial direction is somewhat less straight-forward since shell spacing is not uniform and derivatives on the shell boundaries as well as at the cell centers are required. When the value of a quantity that is known on shell boundaries must be calculated at a shell center, simple averaging is used. When the value of a quantity that is known in shell centers must be calculated on shell boundaries a weighted average is used. Two variables are defined as:

$$R1 = \frac{(\Delta R)_{s+1}}{[(\Delta R)_s + (\Delta R)_{s+1}]} \qquad R2 = \frac{(\Delta R)_s}{[(\Delta R)_s + (\Delta R)_{s+1}]} \qquad (A.4)$$

Here $(\Delta R)_s$ is the radial grid spacing of the s'th shell. The value of Q on the s'th shell is given by $Q_{s+1/2,x}$

$$Q_{s+1/2,x} = R1 \cdot Q_{s,x} + R2 \cdot Q_{s+1,x} \qquad (A.5)$$

When Q is known in shell centers and $\frac{\partial Q}{\partial s}$ is required on a shell boundary, the necessary differencing is straight forward.

$$\left(\frac{\partial Q}{\partial s}\right)_{s+1/2} = \frac{2 \cdot (Q_{s+1,x} - Q_{s,x})}{\phi_s + \phi_{s+1}} \tag{A.6}$$

Here ϕ_s is the dimensionless value of magnetic flux in the s'th shell, and is equal to Δs across the s'th shell. ϕ_s is specified by initial conditions and is independent of x and t in the flux coordinate model. For cases where Q is known at shell centers and $\frac{\partial Q}{\partial s}$ is required at shell centers, the derivative is first found on the boundary and then averages taken to find values in cell centers. The quantity $(1/\partial_s R)$ appears in several expressions in Section 2.4.5. This is easily evaluated from equation (2.27).

$$\frac{1}{\partial_s R} = 2\ RB \tag{A.7}$$

Equations which do not involve diffusion terms (second order spatial derivatives) are evaluated using a straight forward explicit differencing scheme. That is RHS of (A.1) is evaluated using differences of the types described above and then the value of the desired quantity is found at the new timestep through applications of equation (A.2).

The use of an explicit formation on the equations that have diffusion terms would introduce a very restrictive stability condition on the program. If an equation of the form:

$$\frac{\partial}{\partial t} Q = \sigma \frac{\partial^2}{\partial x^2} Q$$

is solved using explicit differencing, the timestep size in the difference equations must satisfy (see Richtmyer and Morton[4])

$$\Delta t \le \frac{\Delta x^2}{2\sigma}$$

This condition would be very restrictive in the present model and render it essentially useless. The use of an implicit formalism eliminates this stability condition. The implicit formalism used in the present routine is as described by Richtmyer and Morton[4]. When equations for a variable Q involve a diffusion term in the x direction the term on the RHS of (A.2) is of the form:

$$\text{RHS} = a\, Q_{s,x-1} + b\, Q_{s,x} + c\, Q_{s,x+1} + \text{other terms}$$

where the coefficients a, b, c may be spatially dependent and may also depend on other variables. Thus equation (A.2) can be expressed as:

$$Q'_{s,x} = Q^{*}_{s,x} + \delta t\{a\, Q_{s,x-1} + b\, Q_{s,x} + c\, Q_{s,x+1} + \text{other terms}\}$$

In an explicit formalism the terms $Q_{s,x}$ in brackets would be the same as $Q^{*}_{s,x}$ in the first step of the two-step scheme. In the second step of the two step scheme, these as well as (other terms) would be based on the results of the first step.

In the implicit formalism used in the present routine however the former equation is re-written as:

$$Q'_{s,x} = Q^{*}_{s,x} + \delta t\{a\, Q^{1/2}_{s,x-1} + b\, Q^{1/2}_{s,x} + c\, Q^{1/2}_{s,x+1} + \text{other terms}\}$$

where

$$Q^{1/2}_{s,x} = \frac{1}{2} (Q^{\circ}_{s,x} + Q'_{s,x})$$

Here the value of $Q_{s,x}$ as found in the first step is not directly used to advance the equation in the second step, however it may be required for transport coefficients and for other equations. The results from the first step are used in the evaluation of other terms in the second step.

The above equation can be re-written in a form:

$$- A_{s,x} Q'_{s,x+1} + B_{s,x} Q'_{s,x} - C_{s,x} Q'_{s,x-1} = D_{s,x}$$

where

$$A_{s,x} = \frac{a}{2} \delta t \quad ,$$

$$B_{s,x} = 1 - \frac{b}{2} \delta t$$

$$C_{s,x} = \frac{c}{2} \delta t$$

$$D_{s,x} = Q^{\circ}_{s,x} + \frac{1}{2} \delta t \{a Q^{\circ}_{s,x-1} + b Q^{\circ}_{s,x} + c Q^{\circ}_{s,x+1} + \text{other terms}\}$$

The solution of this system of equations and the application of boundary conditions followed the procedures outlined by Richtmyer and Morton[4].

A completely analogous technique can be employed to solve equations that contain a diffusion term in the radial direction. However, a somewhat more complex technique must be employed to solve the electron temperature equation which has diffusion terms in both the radial and axial directions. The ion temperature equation has been simplified by neglecting thermal conduction in the axial direction.

One method of numerically solving partial differential equations that have diffusion terms in two directions is through the use of Alternating Direction Implicit (ADI) techniques. An example of the use of such a technique can be found in Lindemuth and Killeen[36]. Basically this technique works by setting up two levels of difference equations. In one level the equation is advanced implicitly in one direction and in the other level it is advanced implicitly in the other direction. Then one timestep uses one set of equations and the next uses the other. It is not clear how such a scheme could be used in the present two step scheme and so another closely related technique was used to advance the electron temperature equation. An equation of this form can be written in the simplified form:

$$\dot{Q} = RD + AD + OT$$

where RD represents terms in RHS of (A.1) that involve radial diffusion terms (second order derivatives in the radial direction), AD represents terms that involve axial diffusion terms (second order derivatives in the axial direction), and OT represents other terms. The above equation then split into two parts:

$$(\dot{Q1}) = RD$$

$$\dot{Q} = (\dot{Q1}) + AD + OT$$

A solution for $(\dot{Q}1)$ is found by using implicit techniques and this solution is inserted into the second equation replacing RD. The solution for $(\dot{Q}1)$ involves advancing Q a timestep assuming radial diffusion terms are the only terms on the RHS.

A.3 Laser Heating

The equations that govern the propagation and absorption of the laser beam are given in dimensionless units in Section 2.3.5. The MHD equations are solved on a computational grid with NX axial points. The laser beam is assumed to be input at the end of the solenoid specified by the NX'th axial point. Laser beam power, specified in dimensionless units P(x), is calculated on axial grid boundaries, and the power absorbed per unit length, E(x) is calculated in cell centers. The indexing of P_x and E_x relative to the indexing of MHD quantities is illustrated in Figure A.3.

The routine contains the option of two types of axial boundary conditions at the point x=1.

(1) Boundary conditions are the same as those at the other end of the solenoid. This boundary condition allows for free streaming of plasma out the end.

(2) Symmetric boundary conditions. This boundary condition assumes that the solution is symmetric about the point x=2. i.e. the point x=2 represents solenoid center and by symmetry arguments, the solution is required only for half of the solenoid.

A more detailed description of these boundary conditonns can be found in Section A.4. When symmetric boundary conditions are employed the beam is assumed to fold over on itself around the point x=2. This is because in a symmetric problem a beam is assumed to be injected from both ends of the solenoid.

The total cross-sectional laser power is given on the axial grid points as P_x. The relative intensity of the beam in each shell is specified by an array $(PR)_{s,x}$ which is used to define the radial profile of the beam (Gaussian for examples in this thesis) and allows for an axial dependence as was required in Chapter 4.

Using the laser absorption coefficient given by Johnston and Dawson[15] and results of Section 2.3.5, k_a is given in dimensionless units as:

$$k_a = K \frac{N^2}{T_e^{3/2}} \ln \Lambda$$

where

$$K = \frac{9.74 \times 10^{-36} N_o^2}{T_o^{3/2}} L_o$$

As described in Section 2.1.5 the absorption coefficient was obtained by averaging over the radial profile. A weighted average is used with the weight at each radial position proportional to the beam intensity at that point. In the finite difference scheme the averaging is over discrete shells and weighting is proportional to total beam power in each shell. The beam power in each shell is proportional to $[(PR)_{s,x} \cdot A_{s,x}]$, thus:

$$\bar{k}_a = \frac{K}{\xi_x} \sum_s \frac{N^2_{s,x}}{(TE)_{s,x}^{3/2}} \ln \Lambda \, (PR)_{s,x} \cdot A_{s,x}$$

where

$$\xi_x = \sum_s (PR)_{s,x} \cdot A_{s,x}$$

The propagation of the laser beam is described by:

$$\frac{1}{c}\frac{\partial}{\partial t} P + \frac{\partial}{\partial x} P = - \bar{k}_a P \qquad (2.49)$$

Normal differencing techniques described earlier in this appendix can be employed to solve this equation, however it is found that the sharp discontinuities expected in the bleaching front lead to great difficulties for this case. There is also a problem in the propagation of a sharp wavefront through a semi-transparent plasma. Instead the following approach is employed. The timestep size is restricted to:

$$\Delta\tau = \frac{m\Delta x}{c}$$

where m is an even integer. $\Delta\tau$ is not the same size as Δt, the timestep size for hydrodynamic quantities, however m is continually adjusted so the time in laser calculations never differs from the time in MHD calculations by more than $\frac{1}{2}\frac{\Delta x}{c}$. With this definition the beam, propagating at a speed c, propagates an integer number of space-step in each timestep $\Delta\tau$.

The beam profile P_x is calculated by advancing it by m small timesteps of size $\frac{\Delta\tau}{m}$ for each full timestep of the main routine and by $\frac{m}{2}$ small timestep for each half step of the main routine. The laser power P_x is advanced in each of the small timesteps through the solution of:

$$\frac{P'_x - P^\circ_{x-1}}{\Delta x} = -\frac{\bar{k}_a}{2} \{P'_x + P^\circ_{x-1}\}$$

or

$$P'_x = P^\circ_{x-1} \frac{[1 - \frac{1}{2}\bar{k}_a \ \Delta x]}{[1 + \frac{1}{2}\bar{k}_a \ \Delta x]}$$

where P'_x represents the new value of P_x and P°_x represents the old value.

The laser power that is absorbed per unit length by the plasma, E, is defined in such a way that energy is conserved in spite of numerical errors which may be made in the first two or three shells of the beam front. E is defined as:

$$E = \bar{k}_a P$$

In each of the m small timesteps an E^i_x is defined as:

$$E^i_x = (P^{i-1}_x - P^i_{x+1})/\Delta x$$

E_x for a full timestep is then given by:

$$E_x = \frac{1}{m} \sum_{i=1}^{m} E^i_x$$

Now the energy absorption rate per particle $(\varepsilon_L)_{s,x}$ must be defined. The total power absorbed in a shell is proportional to both $N_{s,x} \cdot A_{s,x} \cdot (\varepsilon_L)_{s,x}$ and $[N^2_{s,x}/(TE)^{3/2}_{s,x}] \cdot (PR)_{s,x} \cdot A_{s,x} \cdot \ln \Lambda$. Also:

$$E_x = \sum_s [(\varepsilon_L)_{s,x} \cdot N_{s,x} \cdot A_{s,x}]$$

All of these conditions can be satisfied only if:

$$N_{s,x} \cdot A_{s,x} \cdot (\varepsilon_L)_{s,x} = \frac{\left(\frac{N^2_{s,x}}{TE^{3/2}_{s,x}}\right) PR_{s,x} A_{s,x}}{\sum_s \left(\frac{N_{s,x}}{TE^{3/2}_{s,x}}\right) PR_{s,x} A_{s,x}} E_x$$

The denominator of this equation can be rewritten as, $\frac{\xi_x \bar{k}_a}{K}$ so:

$$(\varepsilon_L)_{s,x} = \frac{K}{\xi_x (\bar{k}_a)_x} \frac{N_{s,x} \cdot (PR)_{s,x} \cdot E_x}{(TE)^{3/2}_{s,x}}$$

A.4 A Users Manual for MHD-2D Routine

In this section some of the various features and options incorporated into the computer routine are described. An explanation of the parameters that are specified through the main input file are given. A listing of the input file, which contains a brief description of each of the parameters, can be found at the end of this section.

A.4.1 Start, Stop and Restart

When a simulation is started at a time of t=0 the parameter NTIME must be set to 0. The simulation will then run until either NTOTAL timesteps have been taken, or the simulation time reaches TMAX seconds. If NTIME $\neq$ 0 the routine attempts to read input data from Fortran unit #2 so that it can start from the NTIME'th timestep of a previous simulation. Fortran unit #2 must be attached to a file or device which contains the unformatted output of a previous simulation.

A.4.2 Initial Conditions

If the simulation is started at a time of t=0 the initial conditions are calculated from various parameters input through the input file. If the routine is being restarted from a previous simulation these parameters are ignored and the initial conditions are read directly from Fortran unit #2 as explained in Section A.4.1.

The computational grid used for the numerical calculation consists of NX x-points and M shells. The spacing of the x-points is always uniform however the shell spacing can vary. The initial shell spacing is specified through the parameters R01 and W. The first shell has a radius of R01 cm and the width of the second shell is W cm. The width of the i'th shell is a^{i-1} W where a is calculated by the routine to be such as to make the outer radius of the last shell equal to the specified radius of the solenoid. The solenoid is LO cm long and has a radius of RO. These values of LO and RO are also used as the basic units of length and radius for the transformation to dimensionless variables. The other two basic units needed for the transformation to dimensionless variables (see Section 2.2.3), are given by NO cm^{-3} and TO eV. If the initial density and temperature profile is to be uniform, the plasma will have an initial density of NO cm^{-3}, an initial temperature of TINT eV and an initial magnetic field of BINT Gauss.

The density and temperature can be given an initial profile. The density is specified through parameters NO, R1N, and R2N as:

$$N(r,\ t=0) = \frac{NO}{1 + \exp[(r - R1N)/R2N]}\ \text{cm}^{-3}$$

R1N, and R2N are input in units of cm. The temperature profile is specified through parameters TINT, R1T, and R2T as:

$$T_{e,i}(r,\ t=0) = \frac{TINT}{1 + \exp[(r - R1T)/R2T]}\ \text{eV}$$

A uniform density and temperature profile in the radial direction is specified by setting R1N >> RO, R2N << RO, R1T >> RO and R2T << RO. The magnetic field is set to BINT Gauss at r = RO and is adjusted for r < RO so that radial pressure balance exists.

An initial density profile can be specified in the axial direction through parameters L1, L2, L4, and L5. The initial axial density profile is given by:

$$N(s, t{=}0) = \left(\frac{.999 \cdot NO}{1 + \exp[(s - L1)/L2]} + .001 \right) \cdot \left(\frac{.999 \cdot NO}{1 + \exp[(LO - s - L4)/L5]} + .001 \right)$$

L1, L2, L4 and L5 are input in units of cm. Initial density profiles such as those used in Chapter 4 can be specified with these parameters. A uniform axial profile can be specified by setting L1 = - LO, L4 = - LO, L2 << LO, and L5 << LO.

A.4.3 Output

The main output from the routine is written in unformatted form on Fortran unit #3. It is intended that this unit be attached to a disk file or a magnetic tape. This output consists of the calculated hydrodynamic and laser quantities as well as various other parameters that may be needed to restart the simulation or by other post-processor routines. This data is output after the initial conditions are calculated and after every subsequent NTAPE'th timestep. The output is suppressed if NTAPE = 999.

Formatted output intended for a printer is written on Fortran unit #6. This output consists of the hydrodynamic and laser variables written in dimensionless form. The output is in matrix form and provision is made for only part of the information to be printed. This is controlled by parameters NXI, NXF, M1, MF and MS. Output is from x-points specified by NXI $\leq$ x $\leq$ NXF and from every MS'th shell for M1 $\leq$ s $\leq$ MF. This data is output after initial conditions are calculated, and after every subsequent NPRT'th timestep. Output is suppressed if

NPRT = 999.

A.4.4 Timestep Control

The timestep size is adjusted after every timestep. The timestep size Δt is the maximum allowed by the following four conditions:

1. Δt must be less than a fraction A1 of the maximum allowed by the stability condition. The stability condition is specified by equations (1.1) and (1.2).
2. The electron temperature in any cell cannot change by more than a fraction A2 in a single timestep.
3. Timestep size cannot increase to more than A3 times the previous timestep size.
4. Δt must be less than DTMAX seconds.

The first timestep must only satisfy conditions 1 (with A1 replaced by A11) and 4.

A.4.5 Laser Parameters

The laser beam power as it enters the solenoid is specified through parameters PWR1, PWR2, PWR3, T1, T2, T3, and T4. The beam power is 0 at t = 0, PWR1 at t = T1, PWR2 at t = T2, PWR3 at t = T3 and 0 at t = T4. Power levels between the specified times are calculated by linear interpolation. Power is specified in Watts and time is specified in seconds.

The radial intensity profile is Gaussian as specified by equations (4.1) and (4.2). Parameters needed in equation (4.2) are input as σ_o = RLO, $\frac{1}{2f}$ = BS, and L_3 = L3.

Sometimes it may be desirable to artificially control the bleaching velocity of the laser front. This is accomplished by setting VLAS to the desired bleaching velocity. If no limit is desired VLAS is set to the speed of light.

In Section A.3, an explanation is given of how the laser beam transport equation is advanced in small timesteps for every timestep of the main routine. Under some conditions M can be a very large number and essentially no changes result from limiting M to a smaller number. This can result in significant savings in CPU time. M is limited to being less than MRMAX. Care must be taken when setting this parameter to a small number since it can lead to errors including loss of energy conservation. It is very unlikely for significant errors to result from this feature if MRMAX $\geq$ NX.

A.4.6 Boundary Conditions

Options exist on some boundary conditions as specified in Section 2.5. If TRBC = . TRUE ., the temperature T_e remains a constant at r = RO. If TRBC = . FALSE . $\partial T_e/\partial s = 0$ at r = RO. This is equivalent to assuming a thermally insulated boundary. The thermal pressure P is assumed to satisfy a boundary condition at the solenoid ends of the form $\partial P/\partial x$ = constant. This boundary condition is imposed by assuming the pressure at x-point 0 is (1 - TDP1) times the pressure at x-point 2 and the pressure at x-point (NX+1) is (1-TDP2) times the pressure at x-point (NX-1). If NBOUND = 1 the solenoid is assumed to be open ended and similar boundary conditions are imposed on each end. If NBOUND = 2 symmetric axial boundary conditions are imposed around x-point 2.

A.4.7 Miscellaneous

Artificial viscosity coefficients $\tilde{a}_r$ and $\tilde{a}_x$ defined in Section 2.3, are set to values AVR and AVX. If these quantities are set to values between 1.5 and 2.0 shocks will be spread over several mesh points. Artificial viscosity is turned off by setting these values to zero.

When a laser beam is bleaching its way through a cold plasma there is no MHD motion well ahead of the beam front,and it is just a waste of CPU time to do calculations in this region. Initially, calculations are not done for x-points 1 to NXMIN1. Then as the beam bleaches its way through the plasma more x-points are added to the region in which calculations are made. Calculations are made up to NXMIN x-points ahead of the laser front. This feature is dissabled if NXMIN1 $\leq$ NBOUND.

The ratio of ion mass to proton mass is specified by parameter RM. A deuterium plasma can be simulated by setting RM = 2. Before the program can be used for higher z gasses the transport co-efficients must be modified to include the z dependence.

Calculations on laser propagation and heating are made only for simulation time t < T4. Thus, if the simulation is to be run with no laser heating, setting T4 to a value less than 0 will save CPU time.

A.4.8 Post Processor Routines

Readable output from the main routine is very limited. Additional information can be obtained from other routines which read the unformatted data from the main routine and do additional processing.

Routine TAPRINT reads unformatted data from disk or tape and prints results from pre-selected timesteps.

Routine ENERGY checks validity of the results of a simulation by calculating the extent to which energy and mass were conserved during the simulation. A more complete description of this routine can be found in the next section.

Routine MPLT3D reads unformatted data from disk or tape and creates three dimensional plots of T_e, T_i, v_z, v_r, B, and n_o as well as a plot of the radius of shell boundaries as a function of x. All variables are converted to natural units and scaling is done automatically. Features such as perspecive plotting and removal of hidden lines are included. The routine works interactively on a Tektronix 4013 or 4015 terminal or will create Calcomp files. The routine automatically determines if output is to a Tektronix terminal or a Calcomp file. If it is writing on a terminal each plot is scaled to fill the screen. By pressing return on the terminal the plot is erased and the next one is drawn. If output is to a Calcomp file, plots are automatically spaced to use a minimum amount of plotter paper.

Routine MPLT1D reads unformatted data from disk or tape, and creates the following plots at specified times; (1) laser beam power and axial beam intensity as a function of x; (2) axial values of T_e, T_i, and n_o as a function of x; (3) values of T_e, T_i, and n_o at a given x as a function of r; and (4) the axial values of v_z are plotted as a function of x. In addition, the axial values of T_e, T_i, and n_o at a specified x are plotted as a function of simulation time. In the same manner as MPLT3D, the routine determines if the output is to a Tektronix terminal or a Calcomp file and adjusts its format appropriately.

A.4.8 Listing of Input File

NTIME Initial timestep. If NTIME=0 program starts at t=0. If NTIME=n program starts from n'th timestep. Initial conditions are read off Fortran Unit #2.

NTOTAL Maximum number of timesteps to take. Program will take (NTOTAL-NTIME) timesteps unless t=TMAX first.

TMAX Maximum time (in seconds) for calculation to run.

NPRT Print output frequency; - on Fortran Unit #6. No output if NPRT=999.

NTAPE Unformatted output frequency on Fortran Unit #3. Used for plotting, energy conservation check and restarting program. No output if NTAPE=999.

PWRi Laser powers in Watts at times Ti. Values are calculated by linear interpolation between values of Ti. Laser power is 0.0 at time=0, PWR1 at T1, PWR2 at T2, PWR3 at T3, and 0.0 at time=T4.

Ti See PWRi.

VLAS Bleaching velocity of laser in cm/sec. Value is artificially limited to account for energy used in plasma formation. If no limit is desired set VLAS=3.0D+10.

PP Parameter for averaging on half timesteps. Quantity Q given by Q(S,X)=PP*(Q2(S,XM)+Q2(S,XP))+QQ*Q2(S,X)+DT1*QDOT where QQ=1.0-2.0*PP.

AVR Dimensionless constant multiple for artificial viscosity in the radial direction. (Good value between 1.5 and 2.0).

AVX Dimensionless constant multiple for artificial viscosity in the axial direction. (Good value between 1.5 and 2.0).

A11 Fraction of stability condition that first timestep must satisfy.

A1 Fraction of stability condition that other timesteps must satisfy.

A2 Max. fraction electron temperature in central shells can increase by in a single timestep.

A3 Max. fraction timestep can increase by in a single timestep.

L3 Distance from input end of solenoid (in cm) that laser is focussed.

RLO Min. beam radius. (at position L3).

BS Beam slope. i.e. rate of convergence or divergence of beam.

DTMAX Max. timestep size that is allowed. (in seconds)

NXMIN Number of x points in front of beam front that calculations are to be made.

MHMAX Maximum number of timesteps to be taken in LHEAT per main timestep. Small values save CPU time but can lead to numerical error including loss of energy conservation. Small values are good if the laser beam profile does not change much in a single timestep. There is only need for this to be greater than NX in extreeme cases such as a pulse travelling through a plasma that is optically thin before laser heating.

TRBC — If TRBC=.TRUE., Temperature on Radial Boundary is Constant. If TRBC=.FALSE., Temperature eqn. assumes thermal insulation on radial boundary.

RM — Ratio of (ion mass)/(proton mass). (i.e. RM=2 for deuterium)

TDP1 — Pressure at x-point 0 is assumed to be (1.0-TDP1) times pressure at x-point 2.

TDP2 — Pressure at x-point (NX+1) is assumed to be (1.0-TDP2) times pressure at x-point (NX-1).

NXI — Initial x-point at which data should be printed. Initial output for X=XMIN if NXI=999.

NXF — Final x-point at which data should be printed. Final output for X=NX if NXF=999.

N1 — Initial shell for which output is printed.

NF — Final shell for which output is printed. Final shell for S=N if NF=999.

NS — Every NS'th shell is output on printer.

NBOUND — Type of boundary condition applied to L.H.S. Independent if NBOUND=1, Symetric if NBOUND=2.

NX — Number of x - points.

N — Number of shells.

L0 — Length of solenoid in cm.

R0 — Radius of solenoid in cm.

N0 — Initial plasma density and value used in normalization; in particles/cm**3.

T0 — Value used in normalization of plasma temperature in eV.

R01 & W — R01 is the radius of the first shell. W is the width of the second shell in cm. Width of other shells increase linearly in such a way as to make radius of last shell R0.

BINT — Initial value of B - field in Gauss.

TINT — Initial values for electron and ion temperatures in eV.

R1T — Radius at which temperature decreases to 1/2 of its value in the initial conditions.

R2T — Scale length over which temperature decreases.

R1N — Radius at which density decreases to 1/2 of its value in the initial conditions.

R2N Scale length over which density decreases.

L1 Distance from laser input end of solenoid that initial density increases to 1/2 of its full value.

L2 Scale length over which density increases.

L4 Distance from other end of solenoid that initial density increases to 1/2 of its full value.

L5 Scale length over which density increases.

NXMIN1 Initial value for the number of z-points NOT to calculate. If this value is less than or equal to NBOUND this feature is dissabled.

RDN Radial Dynamics. If RDN=.TRUE. full radial dynamics are calculated. If RDN=.FALSE. Radial Pressure Balance is assumed. This can save CPU time since the stability condition is much less severe so larger timesteps can be taken. If something other than radial hydrodynamic stability is the limiting factor on the timestep size this will NOT lead to significant savings.

```
&INPUT NTIME=0 &END
&INPUT NTOTAL=1000, TMAX=1.5D-06, NPRT=20, NTAPE=10,
PWR1=1.0D+09, PWR2=1.0D+09, PWR3=1.0D+09, T1=10.0D-9, T2=40.0D-09, T3=2.0D-6,
T4=2.2D-6, VLAS=3.0D+10, PP=0.0, AVR=0.0, AVX=0.0,
A11=0.01, A1=0.4, A2=0.5, A3=1.2, L3=25.0, ALO=0.15, BS=0.0
DTMAX=1.0D-08, NXMIN=6, NRMAX=40, TRBC=.FALSE., RN=1.0,
TDP1=0.25, TDP2=0.25, NXI=999, NXP=999, N1=1, NP=999, NS=2
&END
&INPUT NBOUND=1, NX=30, N=15, L0=100.0, R0=1.5, N0=3.E17,
T0=1.0D02, R01=0.045D0, Z=0.045D0, BINT=1.0D+05, TINT=1.0,
R1T=10.0, R2T=0.1, R1N=10.0, R2N=0.15,
L1=-100.0, L2=10.0, L4=-100.0, L5=10.0, NXMIN1=25, RDN=.FALSE.
&END
```

A.5 Conservation of Energy and Mass

The routine ENERGY reads unformatted data from disk or tape, and calculates the total energy and mass of the plasma. After every I'th input record (where I is an input parameter) the following data is printed; (1) total plasma energy; (2) total plasma mass; (3) the amount the plasma energy has changed from previous timestep; (4) the amount of energy that has escaped through solenoid ends; (5) the amount the plasma mass has changed from previous timestep; (6) the amount of mass that has escaped through solenoid ends; (7) the difference between (3) and (4) divided by total plasma energy; and (8) the difference between (5) and (6) divided by total plasma mass.

Energy and mass are calculated and printed out in dimensionless units. A unit of energy is given by $E_o = N_o k_B T_o L_o A_o$ and a unit of mass is $M_o = A_o L_o M_1 N_o$. Two parameters, N1 and N4, specify the axial region in which mass and energy conservation is checked. The rate at which energy and mass is flowing out of the left end of the solenoid is calculated at the x-point N1 while the rate at which energy and mass are flowing out of the right end of the solenoid is calculated at the x-point N4. The total energy and mass is the sum of that calculated at each x-point in the region specified by $(NI + 1) \leq x \leq (N4 - 1)$ plus half of the energy at x-points N1 and N4. If NBOUND = 2 an axis of symmetry exists around the point x = 2 and there is no energy flow past this point. For this case the routine automatically sets N1 = 2 and calculates energy and mass flow past the point x = N4 only.

The total energy (or mass) is found by summing the energy (or mass) calculated in each of the computational cells in the region of interest. In dimensionless units the mass in a cell specified by $M_{s,x}$ is:

$$M_{s,x} = (NA)_{s,x}\,\Delta x$$

In dimensionless units the energy in a cell is the sum of the following components:

Thermal Energy $= (\gamma-1)^{-1} N_{s,x} \cdot (T_e + T_i)_{s,x} \cdot A_{s,x} \cdot \Delta x$

Magnetic Energy $= (B_{s,x})^2 \cdot A_{s,x} \cdot \Delta x$

Kinetic Energy $= \frac{1}{2} N_{s,x}[v_{s,x}^2 + \epsilon_1^2 (v_r)_{s,x}^2]\, A_{s,x} \cdot \Delta x$

Cross-Sectional Laser Energy $= \tilde{c}^{-1} P_x\, \Delta x$; $(\tilde{c} = c/v_o)$

The total energy U is:

$$U = \Delta x \sum_x \tilde{c}^{-1} P_x \sum_s A_{s,x} \{(\gamma-1)^{-1} N_{s,x} \cdot (T_e+T_i)_{s,x} + B_{s,x}^2 + \frac{1}{2} N_{s,x}[v_{s,x}^2 + \epsilon_1^2 (v_r)_{s,x}^2]\}$$

The mass flow out of a solenoid end in a time Δt is (mass density) $\cdot$ (ΔV) where ΔV is the volume of plasma in a shell that has flowed past the solenoid end. ΔV for a computational cell is given by:

$$(\Delta V)_{s,x} = (Av)_{s,x} \cdot \Delta t$$

In dimensionless units the mass flow past a x-point in a time Δt is given by $(\Delta M)_x$ where:

$$(\Delta M)_x = \Delta t \sum_s (NAv)_{s,x}$$

The energy flow past an x-point can be found by summing the following terms:

$$\text{Thermal Energy} = \sum_s (\gamma-1)^{-1} [N(T_e + T_i)\ Av]_{s,x}\ \Delta t$$

$$\text{Kinetic Energy} = \sum_s \frac{1}{2} [N(v^2 + \epsilon_1^2\ v_r^2)\ Av]_{s,x}\ \Delta t$$

(Pressure) ΔV term = work done in pushing gas out the end of the solenoid:

$$= \sum_s [N(T_e + T_i)\ Av]_{s,x}\ \Delta t$$

$$\text{Thermal Conduction} = -\ \delta \sum_s (A\ T_e^{5/2})_{s,x} [(T_e)_{s,x+1} - (T_e)_{s,x-1}]\ \frac{\Delta t}{2\Delta x}$$

where $\delta = (k_{11}\ T_e^{-5/2})_{s,x}$

Laser Energy Flow = $\frac{1}{2}[P_{nx+1-x}$

if NBOUND = 1, and

$$= \frac{1}{2} [P_{nx+1-x} + P_{nx+2-x} - P_{nx-2+x} - P_{nx-3+x}]$$

if NBOUND = 2.

Summing these terms yields the energy flow past an x-point $(\Delta U)_x$.

$$(\Delta U)_x = \sum_s \{[\frac{\gamma}{\gamma-1} N(T_e+T_i) + \frac{1}{2} N(v^2 + \varepsilon_1^2 v_r^2)] A v\}_{s,x} \Delta t$$

$$- \delta \sum_s (A T_e^{5/2})_{s,x} [(T_e)_{s,x+1} - (T_e)_{s,x-1}] \frac{\Delta t}{2\Delta x}$$

+ Laser Energy Flow.

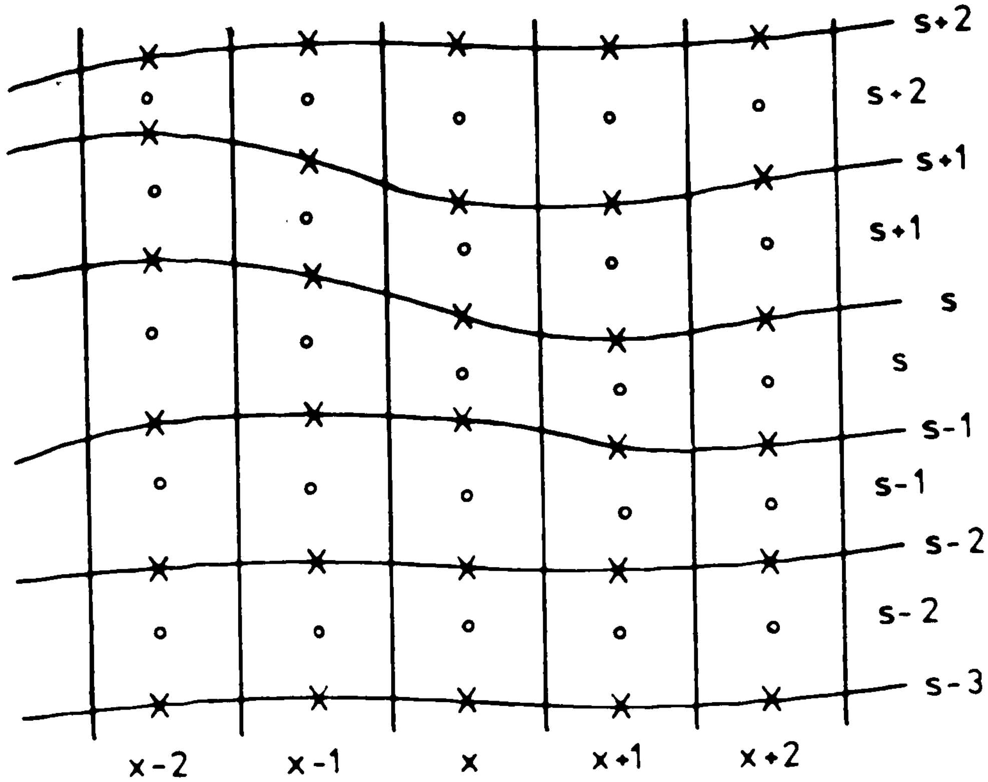

Figure A.1 Spatial positions at which variables are calculated. N, T_e, T_i, v_x, and B are calculated at shell centers denoted by o. R, W, and u are calculated on shell boundaries denoted by x. The indexing of variables is indicated on the bottom and right hand side of the figure.

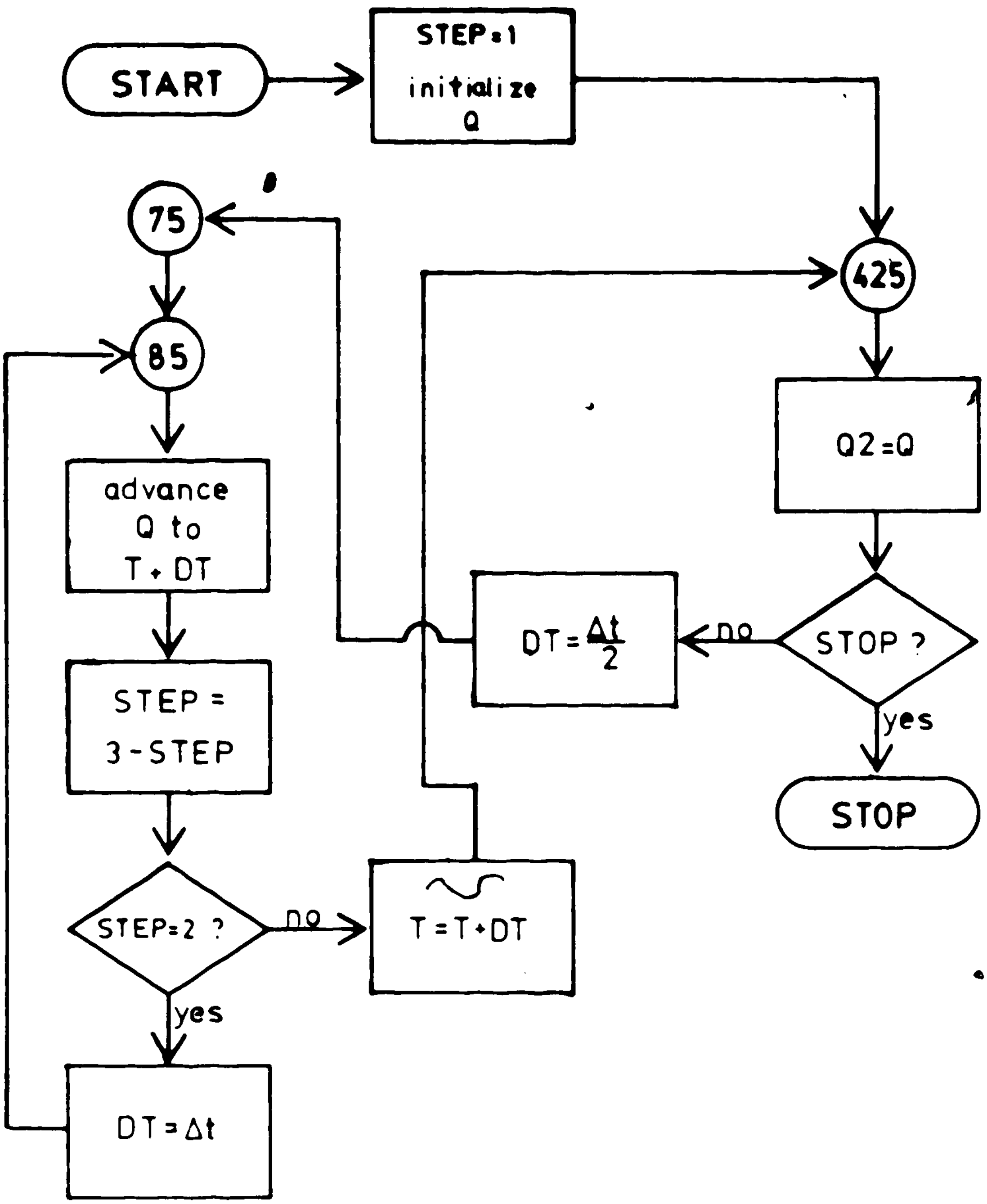

Figure A.2 Flow chart illustrating two-step scheme.

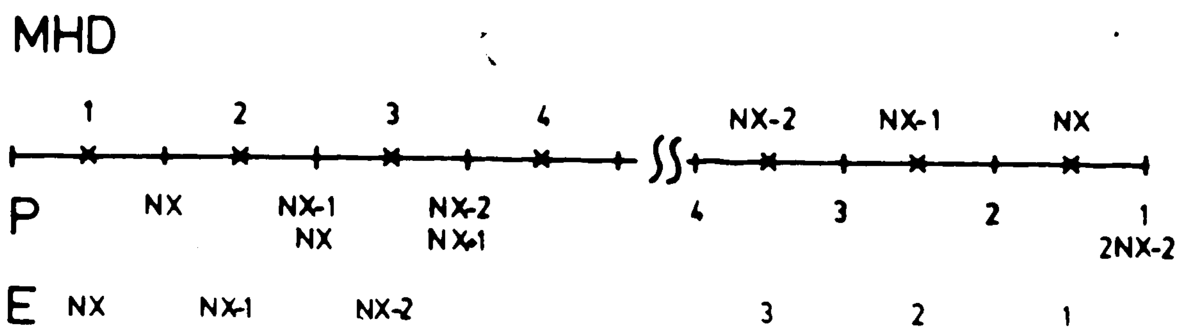

Figure A.3 Position and indexing of numerical grid for calculation of laser power P, and absorbed energy E, relative to positioning and indexing of MHD variables.

Appendix B Linear Susceptibilities

In the following analysis, $F_{T\epsilon}$ (x,t) is the total force acting species ϵ (ϵ = e or i) and $E_{c\epsilon}$ (x,t) is the component of the Coulomb field resulting from density fluctuations of species ϵ. The distribution function $f_\epsilon(x, v, t)$ can be split into two parts; the equilibrium distribution function $f^{o}_\epsilon(v)$, and the perturbation $f'_\epsilon(x, v, t)$.

$$f_\epsilon(x,v,t) = f^{\circ}_\epsilon(v) + f'_\epsilon(x,v,t)$$

The linearized Vlasov equation can be written as:

$$\partial_t f'_\epsilon + v\,\partial_x f'_\epsilon + \frac{F_{T\epsilon}}{m_\epsilon}\,\partial_v f^{\circ}_\epsilon = 0$$

Taking a double Fourier transform in x and t yields:

$$-i\omega\tilde{f}_\epsilon + i\,k\,v\,\tilde{f}_\epsilon = -\frac{F_{T\epsilon}}{m_\epsilon}\,\partial_v f^{\circ}_\epsilon$$

where $\tilde{f}_\epsilon$ and $\tilde{F}_{T\epsilon}$ are the double Fourier transforms of f'_ϵ and $F_{T\epsilon}$ respectively. Thus:

$$\tilde{f}_\epsilon = -\frac{i\tilde{F}_{T\epsilon}}{m_\epsilon(\omega - kv)}\,\partial_v f^{\circ}_\epsilon \qquad \text{(B.1)}$$

From Gauss' Law:

$$\partial_x E_{c\epsilon}(x,t) = 4\pi\, n_\epsilon\, q_\epsilon$$

where q_ε is the species charge and n_ε is the density perturbation defined by:

$$n_\varepsilon(x,t) = n_o \int_{-\infty}^{\infty} f'_\varepsilon(x,v,t)\, dv$$

which implies:

$$\tilde{E}_{c\varepsilon} = \frac{-i\, 4\pi\, n_o\, q_\varepsilon}{k} \int_{-\infty}^{\infty} \tilde{f}_\varepsilon\, dv \qquad (B.2)$$

where $\tilde{E}_{c\varepsilon}(k,\omega)$ is the double Fourier transform of $E_{c\varepsilon}(x,t)$. Substitution of (B.1) into (B.2) yields:

$$\tilde{E}_{c\varepsilon}(\omega,k) = -\frac{1}{q_\varepsilon}\, \chi_\varepsilon(\omega,k)\, \tilde{F}_{T\varepsilon}(\omega,k) \qquad (B.3)$$

where

$$\chi_\varepsilon(\omega,k) = \frac{\omega_{p\varepsilon}^2}{k^2} \int_{-\infty}^{\infty} \frac{k\, \partial_v f^\circ_\varepsilon}{(\omega - kv)}\, dv \qquad (B.4)$$

and $\omega_{p\varepsilon} = (4\pi\, n_o\, q_\varepsilon / m_\varepsilon)^{1/2}$.

Assuming $f^\circ_\varepsilon(v)$ is Maxwellian:

$$f^\circ_\varepsilon(v) = \frac{1}{\sqrt{\pi}\, v_{\theta\varepsilon}} \exp[-\, v^2 / v_{\theta\varepsilon}^2]$$

where $v_{\theta\varepsilon} = (2k_B T_\varepsilon / m_\varepsilon)^{1/2}$ is the thermal velocity of species ε. With this value of f°_ε, χ_ε can be expressed as:

$$\chi_\varepsilon = -\frac{2\omega_{p\varepsilon}^2}{k\sqrt{\pi}\, v_{\theta\varepsilon}^3} \int_{-\infty}^{\infty} \frac{v \exp[-v^2/v_{\theta\varepsilon}^2]}{\omega - kv}\, dv$$

Using techniques discussed in Stix, χ_ε can be re-written as:

$$\chi_\varepsilon(\omega,k) = 2\,\frac{\omega_{p\varepsilon}^2}{k^2} \cdot \frac{1}{v_{\theta\varepsilon}^2}\,\{1 + i\,\alpha_{o\varepsilon}\, F_o^\varepsilon\} \tag{B.5}$$

where $\alpha_{o\varepsilon} = \omega/(kv_{\theta\varepsilon})$ and

$$F_o^\varepsilon = \frac{ik}{\sqrt{\pi}} \int_{-\infty}^{\infty} \frac{\exp[-v^2/v_{\theta\varepsilon}^2]}{(\omega - kv)}\, dv \tag{B.6}$$

Techniques for evaluating F_o^ε are discussed in Stix[31] (Chapter 8).

Appendix C Ponderomotive Force Due to Antiparallel Beams

In this Appendix the ponderomotive force resulting from the mixing of two antiparallel electromagnetic beams is found. Only terms at the beat frequency of the beams are retained.

The force on a particle (electron or ion) is given by

$$\vec{F} = q_\varepsilon(\vec{E} + \frac{1}{c}\vec{v} \times \vec{B}) \qquad \varepsilon = e \text{ or } i \tag{C.1}$$

where q_ε is the species charge, $\vec{E}$ and $\vec{B}$ are the total electric and magnetic fields due to the interacting beams and $\vec{v}$ is the particle velocity. The total vector potential due to the interacting beams can be written as

$$\vec{A} = A_1 e^{i\xi_1} \hat{y} + A_2 e^{i\xi_2} \hat{y} + \text{c.c.}$$

where $\xi_i = (k_i x - \omega_i t)$, $i = 1,2$.

Then to first order the particle velocity is given by

$$m_\varepsilon \frac{\partial \vec{v}}{\partial t} = -\frac{q_\varepsilon}{c} \frac{\partial \vec{A}}{\partial t}$$

or

$$\vec{v} = -\frac{q_\varepsilon \vec{A}}{m_\varepsilon c} \tag{C.2}$$

The ponderomotive force now comes from the second term in (C.1) where v is approximated as in (C.2). This force is given by

$$\vec{F}_\varepsilon = \frac{q_\varepsilon}{c}\,\vec{v} \times \vec{B} = \frac{q_\varepsilon}{c}\,\vec{v} \times (\vec{\nabla} \times \vec{A})$$

or

$$\vec{F}_\varepsilon = -\frac{q_\varepsilon^2}{mc^2}\,(A_1 e^{i\xi_1} + A_2 e^{i\xi_2} + \text{c.c.})(ik_1 A_1 e^{i\xi_1} + ik_2 A_2 e^{i\xi_2} + \text{c.c.})\,\hat{x}$$

Retaining only terms at the beat frequency in this expression leads to

$$\vec{F}_\varepsilon = -\frac{iq_\varepsilon^2}{m_\varepsilon c^2}\,A_1 A_2^* \,k_3 \exp[i(k_3 x - \omega_3 t)]\,\hat{x} + \text{c.c.}$$

where $k_3 = k_1 - k_2$ and $\omega_3 = \omega_1 - \omega_2$.

Appendix D __Listing of Computer Routines__

```
C**********************************************************************
C     *       A TWO DIMENSIONAL MAGNETOHYDRODYNAMIC SIMULATION OF A    *
C     *       PLASMA IN A SOLENOIDAL MAGNETIC FIELD.                   *
C     *        MICHARD D. MILROY & J. N. MCMULLIN   1977.              *
C**********************************************************************
      IMPLICIT REAL(A-H,O-Z)
      REAL MS,MBND,NT32,NDOT,SQRT,LO,NO
      REAL N(30,60),A(30,60),V(30,60),TE(30,60),TI(30,60),
     *   N2(30,60),A2(30,60),V2(30,60),TE2(30,60),TI2(30,60),
     *   AA(30,60),BB(30,60),CC(30,60),DD(30,60),
     *   PHI(31),P(31),G(31),IE(31),II(31),C(31),VDOT(31),
     *   MI(31),ME(31),GM(31),GP(31),ESTAR(31),J(31),
     *   ATEM(31),NTEM(31),VTEM(31),TITEM(31),EP(30,60),KE,KI,
     *   P(120),P2(120),TE15(30,60)
     *   ,TE3(31),TI3(31),DDE(31),DDI(31),AE(31),AI(31),BE(31),BI(31)
     *   ,CE(31),CI(31),DE(31),DI(31),EE(31),EI(31),FE(31),FI(31)
     *   ,PR(30,60),U(30,60),U2(30,60),W(30,60),W2(30,60),UDOT(31)
     *   ,UTEM(31),R(30,60),ATEM1(31),EPS2(31),EM(31),FM(31),MSO,MSM
      INTEGER S,SP,SM,X,XM,XP,STEP,XMIN,XMP1,XMM1
      LOGICAL TRBC,RDN,RPB
      DATA P/120*0.0D0/,P2/120*0.0D0/,STEP/1/,PR/1800*0.0D0/
     1,EP/1800*0.0D0/,RPB/.TRUE./
      COMMON LO,RO,NO,TO,PWR1,PWR2,PWR3,T1,T2,T3,T4,TMAX,VLAS,TDP1,TDP2
     [illegible]AV,AVX,PP1,NX,M,NTOTAL,NBOUND,NTIME,NPRT,NTAPE,NXMIN,RDN,TRBC
C**********************************************************************
C     NTIME = NUMBER OF TIMESTEPS TAKEN SO FAR.                        *
C     NPRT = PRINT OUTPUT FREQUENCY.      NO OUTPUT IF NPRT=999.       *
C     NTAPE = TAPE OUTPUT FREQUENCY.      NO OUTPUT IF NTAPE=999.      *
C     NTOTAL = TOTAL TIMESTEPS TO TAKE.                                *
C**********************************************************************
      CALL INIT(DT,DX,DEL,PCOLL,EPS20,PHI,V,N,A,TE,TI
     1,DELE1,DELE2,DELI1,DELI2,ABO,CS,TIME,TMO,XLAS,R,U,W,XMIN)
      IF (NTIME.NE.0)
     *CALL REST(DT,DX,DEL,PCOLL,EPS20,PHI,V,N,A,TE,TI
     1,DELE1,DELE2,DELI1,DELI2,ABO,CS,P,P2,TIME,TMO,XLAS,R,U,W,XMIN)
      IF(RDN) RPB=.FALSE.
      GAM=5.0D0/3.0D0
      EPS1 = (RO/LO)
      EPS12=(RO/LO)**2
      AVE2 = (AV*EPS1)**2
      MM1=M-1
      NXM1=NX-1
      NXP1=NX+1
      NXP2=NX+2
      HDT=0.5*DT
      TDX=2.0*DX
      NX2=2*NX
```

```
      NX2M1=NX2-1
      NX2M2=NX2M1-1
      NXM1=NX-1
      GM1=GAM-1
      PPP=1.0
      QQQ=2.0-PPP
      P(1)=0.0
      P(M+1)=0.0
      G(1)=0.0
      G(M+1)=0.0
      IE(1)=0.0
      IE(M+1)=0.0
      II(1)=0.0
      II(M+1)=0.0
      UDOT(M) = 0.0
C----------INITIALIZE TEMPORARY ARRAYS
      DO 50 S=1,M
      ATEM(S) = 0.
      NTEM(S) = 0.
      VTEM(S) = 0.
   50 TITEM(S) = 0.
      GOTO425
C----------RETURN POINT OF TIME LOOP
   75 CONTINUE
      TIME = TIME+DT
      NTIME = NTIME+1
      HDT = 0.5*DT
      DT3=1.0D0/(3.0D0*DT)
C------- CALCULATE XMIN.
      IF(XMIN.LT.NBOUND) STOP
      IF(XMIN.GT.NBOUND)
     1     CALL NDPTS(XMIN,P,NX,NBOUND,XXMIN)
      XMP1 = XMIN+1
      XMM1 = XMIN-1
      NPMX = NX+1-XMIN
   85 CONTINUE
      DT1=HDT*STEP
      GDT1=GM1*DT1
      PP=(2-STEP)*PP1
      QQ=1.0D0-2.0D0*PP
C------- DEFINE TE**1.5 AS TE15.
      DO 14 X=1,NX
      DO 14 S=1,M
   14 TE15(S,X) = TE(S,X)*SQRT(TE(S,X))
      IF(TIME.LE.T5)
     1CALL LHEAT(M,NX,STEP,TIME,T1,T2,T3,T4,NBOUND,TE15,XMIN
     2,PWR1,PWR2,PWR3,XLAS,VLAS,RO,ABO,P,P2,PR,L,N,A,TE,EP,DX,NXP1,TO)
C-1--------AXIAL LOOP
      DO 300 X=XMIN,NX
      IBR = 1
      IF(X.EQ.1) IBR = 2
      IF(X.EQ.NX) IBR = 3
      XM = X-1
      XP = X+1
      SUM1 = 0.
      SUM2 = 0.
      ABND = 0.
      NJ1=0.0
      NJ2=0.0
      RSP=0.5*R(1,X)
```

```
      VPS = U(1,X)/N(1,X)
      Q2 = 0.0
      IF(VPS.LT.0.0) Q2 = N(1,X)*ATR2*VPS**2
      PTL = N(1,X)*(TE(1,X)+TI(1,X)) + (PHI(1)/A(1,X))**2
      DNDXL = 0.0
C-2--------FIRST SHELL LOOP
      DO 150 S=1,M
      SP = S+1
      SM = S-1
      NS = N(S,X)
      AS = A(S,X)
      VS = V(S,X)
      TES = TE(S,X)
      TIS = TI(S,X)
      TS = TES + TIS
      PS = NS*TS
      PHS = PHI(S)
      BS = PHS/AS
      WS = W(S,X)
      US = U(S,X)
C-2-1------CALCULATION OF FLUXES THRU UPPER BOUNDARY
      IF(S.EQ.M) GO TO 101
      BSP = PHI(SP)/A(SP,X)
      PH2 = PHI(SP)
      PH12 = PHS+PH2
      ABND = ABND + AS
      RS=RSP
      RSP=0.5*(R(S,X)+R(S+1,X))
      DR1=AS/(2.0*RS)
      DR2=A(S+1,X)/(2.0*RSP)
      DR12=DR1+DR2
      R1=DR2/DR12
      R2=DR1/DR12
      BBND=R1*BS+R2*PH2/A(SP,X)
      DBYB = 4.*ABND*(PH2/A(SP,X)-BS)/PH12
      NBND = R1*NS + R2*N(SP,X)
      VBND = R1*VS + R2*V(SP,X)
      TIBND = R1*TIS + R2*TI(SP,X)
      TEBND = R1*TES + R2*TE(SP,X)
      SQTEBD = SQRT(TEBND)
      EPS2(S) = EPS20/(TEBND*SQTEBD)
      F(SP) = EPS2(S)*NBND*DBYB
      G(SP) = VBND*F(SP)
      FBYN = F(SP)/NBND**GM1
      II(SP) = TIBND*FBYN
      IE(SP) = TEBND*FBYN
      HJ2=16.0*EPS2(S)*ABND*(((PH2/A(S+1,X))-(PHS/AS))/(AS+A(S+1,X)))**2
      GO TO 100
  101 HJ2=0.0
  100 NT32 = NS/TE15(S,X)
      HJUL=0.5*(HJ1+HJ2)
      HJ1=HJ2
      XNU=PCOLL*NT32
      IF(XNU.GT.DT3) XNU=DT3
      TEQ=XNU*(TES-TIS)
C-2-2------CALCULATE SPACE DERIV'S AND NET SHELL FLUXES
      DELF = F(SP)-F(S)
      DELG = G(SP)-G(S)
      PNGM1 = NS**GM1
      DELIE = PNGM1*(IE(SP)-IE(S))
```

```
      DELII = PNGN1*(II(SP)-II(S))
      AN = AS*NS
      GO TO (30,40,20) , IBR
   30 TDN = N(S,XP)-N(S,XM)
      TDV = V(S,XP)-V(S,XM)
      TDTE = TE(S,XP)-TE(S,XM)
      TDTI = TI(S,XP)-TI(S,XM)
      TDP = N(S,XP)*(TE(S,XP)+TI(S,XP))-N(S,XM)*(TE(S,XM)+TI(S,XM))
      TDNAV = +(NS*AS*(V(S,XP)-V(S,XM)) + NS*VS*(A(S,XP)-A(S,XM))
     1         + AS*VS*(N(S,XP)-N(S,XM)))
      ATM = A(S,XM)*TE(S,XM)*TE15(S,XM)
      ATO = AS*TES*TE15(S,X)
      ATP = A(S,XP)*TE(S,XP)*TE15(S,XP)
      TDU = U(S,XP) - U(S,XM)
      TDW = W(S,XP) - W(S,XM)
      VXM=V(S,XM)
      VXP=V(S,XP)
      QP=0.0
      QM=0.0
      IF(VXP.LT.VS) QP=(AVX*(VXP-VS))**2 * 0.5*(NS+N(S,XP))
      IF(VS.LT.VXM) QM=(AVX*(VS-VXM))**2 * 0.5*(N(S,XM)+NS)
      GO TO 21
   20 TDN = 2.0*(N(S,X)-N(S,XM))
      TDV = 2.0*(V(S,X)-V(S,XM))
      TDTE = 2.0*(TE(S,X)-TE(S,XM))
      TDTI = 2.0*(TI(S,X)-TI(S,XM))
      TDP = -N(S,XM)*(TE(S,XM)+TI(S,XM))*TDP2
      TDNAV=2.0D0*(NS*AS*(VS-V(S,XM))+NS*VS*(AS-A(S,XM))
     1+AS*VS*(NS-N(S,XM)))
      ATM = A(S,XM)*TE(S,XM)*TE15(S,XM)
      ATO = AS*TES*TE15(S,X)
      ATP = ATO
      TDU = 2.0*(U(S,X)-U(S,XM))
      TDW = 2.0*(W(S,X)-W(S,XM))
      QP = 0.0
      QM = 0.0
      GO TO 21
   40 TDN = 2.0*(N(S,XP)-N(S,X))
      TDV = 2.0*(V(S,XP)-V(S,X))
      TDTE = 2.0*(TE(S,XP)-TE(S,X))
      TDTI = 2.0*(TI(S,XP)-TI(S,X))
      TDP = N(S,XP)*(TE(S,XP)+TI(S,XP))*TDP1
      TDNAV=2.0D0*(NS*AS*(V(S,XP)-VS)+NS*VS*(A(S,XP)-AS)
     1+AS*VS*(N(S,XP)-NS))
      ATM = AS*TES*TE15(S,X)
      ATO = ATM
      ATP = A(S,XP)*TE(S,XP)*TE15(S,XP)
      TDU = 2.0*(U(S,XP)-U(S,X))
      TDW = 2.0*(W(S,XP)-W(S,X))
      QP = 0.0
      QM = 0.0
   21 CONTINUE
      PTU = NBND*(TEBND+TIBND) + BBND**2
      DWDXU = TDW/TDX
      DPT = PTU - PTL
      DWDXM = 0.5*(DWDXL+DWDXU)
      PTL = PTU
      DWDXL = DWDXU
      ET = DWDXM*DPT/AN
      DQDX = (QP-QM)/DX
```

```
      C(S) = -( TDNAV/TDX + DELP )
      VDOT(S) = -( VS*TDV+TDP/NS )/TDX - (DELG-VS*DELP)/AN - DQDX/NS
     1 + ET
      HE(S) = -VS*(TDTE-GM1*TES*TDN/NS)/TDX - (DELIE-TES*DELP)/AN
     1 -TEQ + GM1*EP(S,X) + GM1*HJUL/NS
      HI(S) = -VS*(TDTI-GM1*TIS*TDN/NS)/TDX - (DELII-TIS*DELP)/AN
     1 +TEQ
      GP = 0.5*GM1*DEL*DT/(NS*AS*DX**2)
      GM(S) = GP*(ATM+ATO)
      GP(S) = GP*(ATP+ATO)
C------- CALCULATE COEFFICIENTS FOR QUASI-IMPLICIT PERP. HEAT FLOW.
      IF(S.EQ.M) GO TO 5
      KE = DELE1*TEBND*TEBND*SQTEBD
     1      / (1.0 + DELE2*TEBND**3*BBND**2/(NBND**2))
      KI = DELI1*TIBND*TIBND*SQRT(TIBND)
     1      / (1.0 + DELI2*TIBND**3*BBND**2/(NBND**2))
      Z=4.0D0*ABND/(AS+A(SP,X))
      DDE(S)=Z*KE
      DDI(S)=Z*KI
      IF(S.EQ.1) GO TO 5
      Z=GDT1/(NS*AS)
      AE(S)=Z*DDE(S)*PPP
      AI(S)=Z*DDI(S)*PPP
      BE(S)=1.0+Z*(DDE(S)+DDE(S-1))*PPP
      BI(S)=1.0+Z*(DDI(S)+DDI(S-1))*PPP
      CE(S)=Z*DDE(S-1)*PPP
      CI(S)=Z*DDI(S-1)*PPP
      DE(S)=TE2(S,X)+Z*(DDE(S)*TE2(S+1,X)-(DDE(S)+DDE(S-1))*TE2(S,X)
     1   +DDE(S-1)*TE2(S-1,X))*QQQ
      DI(S)=TI2(S,X)+Z*(DDI(S)*TI2(S+1,X)-(DDI(S)+DDI(S-1))*TI2(S,X)
     1   +DDI(S-1)*TI2(S-1,X))*QQQ + HI(S)*DT1
    5 CONTINUE
      IF(RPB) GO TO 150
C------- SOLVE EQNS. ASSOCIATED WITH RADIAL DYNAMICS.
C------- FIRST FIND ARTIFICIAL VISCOSITY.
      IF(S.EQ.M) GO TO 150
      VPSP = U(SP,X)/R(SP,X)
      Q1 = 0.0
      IF(VPSP.LT.VPS) Q1 = NS*AVE2*(VPSP-VPS)**2
      DQDS = 2.0*(Q1-Q2)/PH12
      Q2 = Q1
      VPS = VPSP
      IF(S.NE.1) GO TO 161
      DUDS = U(2,X)/PH12
      GO TO 162
  161 DUDS = (U(SP,X)-U(SM,X))/PH12
  162 DUDX = TDU/TDX
      D = 4.0*NS*BBND*(BSP-BS)/PH12
      Z = 2.0*BBND*NS/(EPS12*NBND)
      DPDS = ((N(SP,X)*(TE(SP,X)+TI(SP,X))) - (NS*TS))*2.0/PH12
      DB2DS = ((BSP**2) - (BS**2))*2.0/PH12
      UDOT(S) = -VBND*DUDX - EPS2(S)*D*DUDS + (US**2)/NS
     1      -Z*(DPDS + DB2DS + DQDS)
C------- SOLVE FOR W SEMI-IMPLICITLY.
      DWDX = TDW/TDX
      Z = 2.0*EPS2(S)*(BBND**2)*NS
      IF(S.NE.1) GO TO 163
      AM = DT1*Z/(PH2*PH12)
      BM = 1.0 + DT1*Z*(1.0/(PH2*PH12) + 1.0/(PHS*PH12))
      DM = W2(S,X) + DT1*(2.0*US - VBND*DWDX)
```

```
     1      + AM*W2(SP,X) - (BM-1)*W2(S,X)
      EM(S) = AM/BM
      FM(S) = DM/BM
      GO TO 150
  163 AM = DT1*Z/(PM2*PM12)
      CM = DT1*Z/(PMS*PM12)
      BM = 1.0 + AM + CM
      DM = W2(S,X) + DT1*(2.0*US - TBND*DWDX)
     1      + AM*W2(SP,X) - (BM-1)*W2(S,X) + CM*W2(SM,X)
      EM(S) = AM/(BM - CM*EM(SM))
      FM(S) = (DM + CM*FM(SM))/(BM -CM*EM(SM))
  150 CONTINUE
      IF(RPB) GO TO 102
C------- SOLVE FOR W(S,X).
      W(M,X) = W2(M,X)
      DO 164 J=1,MM1
      S = M-J
      SP = S+1
      W(S,X) = EM(S)*W(SP,X)+FM(S)
  164 CONTINUE
C------- USE NEW VALUE OF W TO FIND ATEM1(S).
      ATEM1(1) = W(1,X)
      DO 165 S=2,M
  165 ATEM1(S) = (W(S,X) - W(S-1,X))
C-------- FIND MORE COEFICIENTS FOR IMPLICIT EQUATIONS.
  102 Z=GDT1/(W(1,X)*A(1,X))
      EEE=Z*DDE(1)*PPP
      FFF=1.0+EEE
      GGG=TE2(1,X)+EEE*(TE2(2,X)-TE2(1,X))*QQQ/PPP
      EE(1)=EEE/FFF
      FE(1)=GGG/FFF
      EEE=Z*DDI(1)*PPP
      FFF=1.0+EEE
      GGG=TI2(1,X)+EEE*(TI2(2,X)-TI2(1,X))*QQQ/PPP + HI(1)*DT1
      EI(1)=EEE/FFF
      FI(1)=GGG/FFF
      DO 6 S=2,MM1
      EE(S)=AE(S)/(BE(S)-CE(S)*EE(S-1))
      EI(S)=AI(S)/(BI(S)-CI(S)*EI(S-1))
      FE(S)=(DE(S)+CE(S)*FE(S-1))/(BE(S)-CE(S)*EE(S-1))
      FI(S)=(DI(S)+CI(S)*FI(S-1))/(BI(S)-CI(S)*EI(S-1))
    6 CONTINUE
C-------- FIND TE3(M) AND TI3(M).
      IF (TRBC) GO TO 15
      Z=GDT1/(W(M,X)*A(M,X))
      FFF=Z*DDE(M-1)*PPP
      EEE=1.0+FFF
      GGG=TE2(M,X)-FFF*(TE2(M,X)-TE2(M-1,X))*QQQ/PPP
      TE3(M)=(GGG+FFF*FE(MM1))/(EEE-FFF*EE(MM1))
      FFF=Z*DDI(MM1)*PPP
      EEE=1.0+FFF
      GGG=TI2(M,X)-FFF*(TI2(M,X)-TI2(MM1,X))*QQQ/PPP + HI(M)*DT1
      TI3(M)=(GGG+FFF*FI(MM1))/(EEE-FFF*EI(MM1))
      GO TO 16
   15 TE3(M) = TE2(M,X)
      TI3(M) = TI2(M,X)
   16 CONTINUE
C------- FIND TE3(S) AND TI3(S)  -  S=MM1 TO 1.
      DO 7 J=1,MM1
      S=M-J
```

```
      TE3(S)=EE(S)*TE3(S+1)+FE(S)
      TI3(S)=EI(S)*TI3(S+1)+FI(S)
    7 CONTINUE
C------- ADD RATE OF CHANGE OF TE AND TI TO HE(S) AND HI(S).
      DO 8 S=1,M
      TEDT=(TE3(S)-TE2(S,X))/DT1
      TIDT=(TI3(S)-TI2(S,X))/DT1
      HE(S)=HE(S)+TEDT
      HI(S)=TIDT
    8 CONTINUE
      IF(RDN) GO TO 103
C-2-4------CALCULATE TERMS FOR ADOT SUMS
      DO 9 S=1,M
      GO TO (34,44,24) , IBR
   34 THETA = (GP(S)*TE(S,XP)-(GM(S)+GP(S))*TE(S,X)+GM(S)*TE(S,XM))/DT
      GO TO 25
   24 THETA = (GP(S)*TE(S,X)-(GM(S)+GP(S))*TE(S,X)+GM(S)*TE(S,XM))/DT
      GO TO 25
   44 THETA = (GP(S)*TE(S,XP)-(GM(S)+GP(S))*TE(S,X)+GM(S)*TE(S,X))/DT
   25 CONTINUE
      HSTAR(S) = N(S,X)*A(S,X)*(HE(S)+HI(S)+THETA)
      PISTAR = 2.*(PHI(S)/A(S,X))**2 + GAM*N(S,X)*(TE(S,X)+TI(S,X))
      SUM1 = SUM1 + (HSTAR(S)+GAM*(TE(S,X)+TI(S,X))*C(S))/PISTAR
      SUM2 = SUM2 + A(S,X)/PISTAR
    9 CONTINUE
      PIDOT = SUM1/SUM2
C-3--------SECOND SHELL LOOP
  103 CONTINUE
      ATOT=0.0
      DO 275 S=1,M
      NS = N(S,X)
      AS = A(S,X)
      VS = V(S,X)
      TES = TE(S,X)
      TIS = TI(S,X)
      TS = TES + TIS
      PS = NS*TS
      PHS = PHI(S)
      BS = PHS/AS
C-3-1------CALCULATE TIME DERIVATIVES
      IF(RPB)
     1ADOT = (HSTAR(S)+GAM*TS*C(S)-AS*PIDOT)/(2.*BS*BS+GAM*PS)
      IF(RDN) ADOT = (ATEM1(S)-A2(S,X))/DT1
      NDOT = (C(S)-NS*ADOT)/AS
      DLNDOT = GM1*NDOT/NS
      TIDOT = HI(S) + TIS*DLNDOT
C-3-2------Q IS NON-CONDUCTIVE PART OF TEDOT
      Q(S) = HE(S) + TES*DLNDOT
C------- TRANSFER TEMP. TO PERM.
      IF(X.EQ.XMIN) GO TO 60
      A(S,XM) = ATEM(S)
      ATOT = ATOT+A(S,XM)
      IF(RPB) W(S,XM) = ATOT
      N(S,XM) = NTEM(S)
      V(S,XM) = VTEM(S)
      TI(S,XM) = TITEM(S)
      IF(RDN) U(S,XM) = UTEM(S)
      R(S,XM) = SQRT(W(S,XM))
C------- ADVANCE TEMP. QUANTITIES TO T+(NDT*STEP).
   60 CONTINUE
```

```
      GO TO (37,47,47) ,IBR
   37 CONTINUE
      IF(RDN) ATEM(S) = ATEM1(S)
      IF(RDN) UTEM(S) = PP*(U2(S,XM)+U2(S,XP)) + QQ*U2(S,X) +DT1*UDOT(S)
      IF(NPB) ATEM(S) = PP*(A2(S,XM)+A2(S,XP))+QQ*A2(S,X) + DT1*ADOT
      NTEM(S) = PP*(N2(S,XM)+N2(S,XP))+QQ*N2(S,X) + DT1*NDOT
      VTEM(S) = PP*(V2(S,XM)+V2(S,XP))+QQ*V2(S,X) + DT1*VDOT(S)
      TITEM(S) = PP*(TI2(S,XM)+TI2(S,XP))+QQ*TI2(S,X) + DT1*TIDOT
      GO TO 27
   47 CONTINUE
      IF(RDN) ATEM(S) = ATEM1(S)
      IF(RDN) UTEM(S) = U2(S,X) + DT1*UDOT(S)
      IF(RPB) ATEM(S) = A2(S,X) + DT1*ADOT
      NTEM(S) = N2(S,X) + DT1*NDOT
      VTEM(S) = V2(S,X) + DT1*VDOT(S)
      TITEM(S) = TI2(S,X) + DT1*TIDOT
   27 CONTINUE
C-3-5------COEFF'S FOR IMPLICIT SCHEME
      GO TO (38,48,28) , IBR
   38 CC(S,X)=+0.25*STEP*GM(S)
      AA(S,X)=+0.25*STEP*GP(S)
      BB(S,X) = 1.+AA(S,X)+CC(S,X)
      DD(S,X) = 0.5*STEP*DT*Q(S)+(PP+CC(S,X))*TE2(S,XM)+(PP+AA(S,X))*
     C TE2(S,XP) +(QQ-CC(S,X)-AA(S,X))*TE2(S,X)
      GO TO 29
   28 AA(S,X)=0.0
      BB(S,X)=1.0+0.25*STEP*GM(S)
      CC(S,X)=0.25*STEP*GM(S)
      DD(S,X) = 0.5*STEP*DT*Q(S)+(CC(S,X))*TE2(S,XM)+(AA(S,X))*
     C TE2(S,X) +(1.0-CC(S,X)-AA(S,X))*TE2(S,X)
      GO TO 29
   48 AA(S,X) = 0.25*STEP*GP(S)
      BB(S,X) = 1.0+AA(S,X)
      CC(S,X) = 0.0
      DD(S,X) = 0.5*STEP*DT*Q(S)+(CC(S,X))*TE2(S,X)+(AA(S,X))*
     C TE2(S,XP) +(1.0-CC(S,X)-AA(S,X))*TE2(S,X)
   29 CONTINUE
  275 CONTINUE
  300 CONTINUE
C-3-6------TRANSFER TEMP TO PERM AT NX
      ATOT = 0.0
      DO 310 S=1,M
      N(S,NX) = NTEM(S)
      A(S,NX) = ATEM(S)
      ATOT = ATOT+A(S,NX)
      IF(RPB) W(S,NX) = ATOT
      V(S,NX) = VTEM(S)
      IF(RDN) U(S,NX) = UTEM(S)
      R(S,NX) = SQRT(W(S,NX))
  310 TI(S,NX) = TITEM(S)
C*********************************************************************
C     *                  IMPLICIT EQUATIONS                          *
C     *  AS SOLVED BY RICHTMYER AND MORGAN,  -  PAGE 198.            *
C     *  T(S,X)=E(S,X)*T(S,X+1)+F(S,X)                               *
C     *  SET E(S,X) TO AA(S,X) AND F(S,X) TO BB(S,X)                 *
C*********************************************************************
      M1 = M
      IF (TRBC) M1 = M-1
      GO TO (52,51) , NBOUND
   51 CONTINUE
```

```
C---------------- CASE WITH SYMETRIC BOUNDARY CONDITIONS.
      IF (XMIN.EQ.2) GO TO 2
      DO 1 S=1,N
      AA(S,XMIN)  =  AA(S,XMIN)/BB(S,XMIN)
    1 BB(S,XMIN)  =  (DD(S,XMIN)+CC(S,XMIN)*TE2(S,XMIN))/BB(S,XMIN)
      GO TO 3
    2 DO 340 S=1,N
      AA(S,2)=(AA(S,2)+CC(S,2))/BB(S,2)
  340 BB(S,2)=DD(S,2)/BB(S,2)
    3 DO 350 X=XMP1,NXM1
      DO 350 S=1,N
      AA(S,X)=AA(S,X)/(BB(S,X)-CC(S,X)*AA(S,X-1))
  350 BB(S,X)=(DD(S,X)+CC(S,X)*BB(S,X-1))/(BB(S,X)-CC(S,X)*AA(S,X-1))
C-2---------APPLY RIGHT B.C.
      DO 370 S=1,N1
  370 TE(S,NX)  =  (DD(S,NX)+CC(S,NX)*BB(S,NX-1))/(BB(S,NX)-CC(S,NX)
     1              * AA(S,NX-1))
      DO 400 J=2,NPMX
      X=NX-J+1
      DO 400 S=1,N1
  400 TE(S,X)=AA(S,X)*TE(S,X+1)+BB(S,X)
C-4---------APPLY LEFT B.C.
      IF(XMIN.GT.2) GO TO 65
      DO 410 S=1,N
      A(S,1)  =  A(S,3)
      N(S,1)  =  N(S,3)
      V(S,1)  =  -V(S,3)
      U(S,1)  =  U(S,3)
      W(S,1)  =  W(S,3)
      R(S,1)  =  R(S,3)
      TE(S,1)  =  TE(S,3)
  410 TI(S,1)  =  TI(S,3)
      GO TO 65
   52 CONTINUE
C--------------------- CASE WITH EACH END TREATED INDIVIDUALLY.
C--------------------- APPLY LEFT B.C.
      IF(XMIN.EQ.1) GO TO 12
      DO 11 S=1,N
      AA(S,XMIN)  =  AA(S,XMIN)/BB(S,XMIN)
   11 BB(S,XMIN)  =  (DD(S,XMIN)-CC(S,XMIN)*TE2(S,XMIN))/BB(S,XMIN)
      GO TO 13
   12 DO 341 S=1,N
      AA(S,1)  =  AA(S,1)/BB(S,1)
  341 BB(S,1)  =  DD(S,1)/BB(S,1)
   13 DO 351 X=XMP1,NXM1
      DO 351 S=1,N
      AA(S,X)=AA(S,X)/(BB(S,X)-CC(S,X)*AA(S,X-1))
  351 BB(S,X)=(DD(S,X)+CC(S,X)*BB(S,X-1))/(BB(S,X)-CC(S,X)*AA(S,X-1))
C-2---------APPLY RIGHT B.C.
      DO 371 S=1,N1
  371 TE(S,NX)  =  (DD(S,NX)+CC(S,NX)*BB(S,NX-1))/(BB(S,NX)-CC(S,NX)
     1              * AA(S,NX-1))
      DO 401 J=2,NPMX
      X=NX-J+1
      DO 401 S=1,N1
  401 TE(S,X)=AA(S,X)*TE(S,X+1)+BB(S,X)
C----------CHANGE STEP
   65 STEP = 3-STEP
      IF(STEP.EQ.2) GOTO 85
C-------- ADJUST TIMESTEP SIZE IF NECESSARY.
```

```
      CALL DTIME(DT,NX,M,TE,TI,TE2,N,PHI,A,R,EPS1,XMIN,RDN,DX)
C----------COPY NEW SOLUTION
  425 CONTINUE
      DO 450 X=1,NX
      DO 450 S=1,M
      A2(S,X) = A(S,X)
      N2(S,X) = N(S,X)
      V2(S,X) = V(S,X)
      U2(S,X) = U(S,X)
      W2(S,X) = W(S,X)
      TE2(S,X) = TE(S,X)
  450 TI2(S,X) = TI(S,X)
C----------OUTPUT ROUTINE
      IF(((NPRT*(NTIME/NPRT)).EQ.NTIME).AND.(NPRT.NE.999))
     1CALL RITE(NTIME,TIME,M,NX,A,TE,TI,N,PHI,V,P,THO,DT,U,R,EP,XMIN
     2,RPB)
      IF(((NTAPE*(NTIME/NTAPE)).EQ.NTIME).AND.(NTAPE.NE.999))
     1CALL TPOUT(NTAPE,NTIME,TIME,M,NX,LO,RO,NO,TO,DT,A
     2,TE,TI,N,PHI,V,P,NBOUND,XLAS,U,W,XMIN,RDN,RM)
      IF ((NTIME.GE.NTOTAL) .OR. (TIME.GE.TMAX)) STOP
C----------END OF OUTPUT
      GO TO 75
      END
      SUBROUTINE INIT(DT,DX,DEL,PCOLL,EPS20,PHI,V,N,A,TE,TI
     1,DELE1,DELE2,DELI1,DELI2,ADO,CS,TIME,THO,XLAS,RD,U,WO,XMIN)
C*********************************************************************
C     SUBROUTINE INPUTS INITIAL CONDITIONS.                          *
C               RICHARD D. MILROY        77-04-23                    *
C*********************************************************************
      IMPLICIT REAL(A-H,O-Z)
      REAL PHI(31),V(30,60),N(30,60),A(30,60),TE(30,60),TI(30,60)
     1,NO,LO,KB,MI,LMDA,LLMDA,R(31),PR(30,60),MINT(31),RD(30,60)
     2,U(30,60),WO(30,60),L1,L2,L3,L4,L5,NCRIT
      INTEGER S,X,XMIN
      LOGICAL TRBC,RDN
      COMMON LO,RO,NO,TO,PWR1,PWR2,PWR3,T1,T2,T3,T4,TMAX,VLAS,TDP1,TDP2
     *,RM
     1,TS,AVR,AVX,PP,NX,M,NTOTAL,NBOUND,NTIME,NPRT,NTAPE,NXMIN,RDN,TRBC
     2/TSS/A11,A1,A2,A3,DTMAX/LASP/L3,RLO,BS,NCRIT,TL,DTL,NRMAX
     3/PRT/NXI,NXP,M1,MP,MS
      NAMELIST /INPUT/ NTIME,NTOTAL,NPRT,NTAPE,PR,PWR1,PWR2,PWR3
     1,T1,T2,T3,T4,TMAX,NBOUND,NX,M,LO,RO,NO,TO,RO1,W,VLAS
     2,BINT,TINT,NINT,PP,AVR,A11,A1,A2,A3,L1,L2,L3,L4,L5,RLO,BS,DTMAX
     3,NXMIN,NXMIN1,RDN,TDP1,TDP2,NXI,NXP,M1,MP,MS,AVX,NRMAX
     4,TRBC,R1T,R2T,R1N,R2N,RM
      TIME=0.0D0
      XLAS=0.0
      READ(5,INPUT)
      IF(NTIME.NE.0) GO TO 2
      READ(5,INPUT)
      RM1 = RM
      READ(5,INPUT)
      DO 6 I=1,M
    6 MINT(I) = 1.0
      L1=L1/LO
      L2=L2/LO
      L3=L3/LO
      L4=L4/LO
      L5=L5/LO
      R1T = R1T/RO
```

```
      R2T = R2T/R0
      R1N = R1N/R0
      R2N = R2N/R0
      NCRIT = 9.96D+18/N0
      XMIN = NXM1
      IF(XMIN.LT.NBOUND) XMIN=NBOUND
      RLO=RLO/R0
      BS=BS*L0/B0
      KB=1.6D-12
      C=3.0D+10
      MI=1.67D-24*MM
      P0=N0*KB*T0
      B0=DSQRT(8.0*3.14159*P0)
      V0=DSQRT(KB*T0/MI)
      TM0=L0/V0
      A0=3.14159*R0**2
      PW0=P0*A0*V0*1.0D-07
      EPS1 = (R0/L0)
      CS=C/V0
      LMDA=1.356E+28*((KB*T0)**3/(3.14159*N0))**0.5
      LLMDA=ALOG(LMDA)
      DIL=1.9D21*T0**2.5/(LLMDA*N0*V0*L0)
      DELE1 = (1.9D+21/LLMDA)*(T0**2.5*L0)/(N0*V0*R0**2)
      DELE2 = (2.49D+25/LLMDA**2)*(T0**3*B0**2)/(N0**2)
      DELI1 = (7.8D+19/LLMDA)*(T0**2.5*L0)/(N0*V0*R0**2)*SQRT(AM)
      DELI2 = (7.8D+22/LLMDA**2)*(T0**3*B0**2)/(N0**2)
      SIGMA=6.7D13*T0**1.5/LLMDA
      EPS20=C**2*L0/(6.28319*SIGMA*R0**2*V0)
      PCOLL=3.29D-09*N0*LLMDA/(T0**1.5)*L0/(V0*AM)
      ABO=9.74E-36*N0**2*L0/T0**1.5
      DX=1.0/(NX-NBOUND)
      PWR1 = PWR1/PW0
      PWR2 = PWR2/PW0
      PWR3 = PWR3/PW0
      T1 = T1/TM0
      T2 = T2/TM0
      T3 = T3/TM0
      T4 = T4/TM0
      T5 = T4 + 3.0/CS
      TMAX = TMAX/TM0
      VLAS=VLAS/V0
      R01=R01/R0
      W=W/R0
      BINT=BINT/B0
      TINT=TINT/T0
C-------- SPECIFY SHELL RADII.
      RR=1.0-R01
      AA=(RR/W)**(1.0/(N-2))
      DO 99 I=1,100
   99 AA=(((RR*(AA-1.0))+W)/W)**(1.0/(N-2))
      R(1)=0.0
      R(2)=R01
      R(3)=R01+W
      NP1=N+1
      DO 100 I=4,NP1
  100 R(I)=R(I-1)+AA*(R(I-1)-R(I-2))
C-------- SET INITIAL VALUE FOR DT.
      DTMAX = DTMAX/TM0
C-------- SET INITIAL VALUE OF DT AT A11 = VALUE ALLOWED BY STABILITY.
      C = (2.0*1.6667*TINT) + (2.0*BINT**2/NINT(1))
```

```
      C = SQRT(C)
      DZ = DX
      IF(RDN)  DZ=R01*EPS1
      DT = A11*(DZ/C)
      IF(DT.GT.DTMAX) DT=DTMAX
      DO 1 X=1,NX
      DO 1 S=1,N
      TE(S,X)=TINT
      TI(S,X)=TINT
      N(S,X)=NINT(S)
      A(S,X)=(R(S+1)**2-R(S)**2)
    1 V(S,X)=0.0
      DO 4 X=1,NX
      DO 4 S=1,N
      RO(S,X) = R(S+1)
      U(S,X)=0.0
    4 WO(S,X)=R(S+1)**2
C-------- MAKE RADIAL PROFILE FOR TEMPERATURE AND DENSITY.
      DO 7 S=1,N
    7 R(S)  = 0.5*(R(S)+R(S+1))
      AT1 = 1./(TINT*TO)
      AT2 = 1.0-AT1
      AN1 = 0.001
      AN2 = 1.0-AN1
      DO 8 S=1,N
      ZZT = R(S)  - R1T
      ZZN = R(S)  - R1N
      XNT = AT2/(1.+EXP(ZZT/R2T))  + AT1
      XNN = AN2/(1.+EXP(ZZN/R2N))  + AN1
      DO 8 X=1,NX
      TE(S,X)  = XNT*TE(S,X)
      TI(S,X)  = XNT*TI(S,X)
    8 N(S,X)  = XNN*N(S,X)
      K=NX/2
      CONST=NINT(N)*(TE(N,K)+TI(N,K))+RINT**2
      DO 3 S=1,N
    3 PHI(S)  = SQRT(CONST-N(S,K)*(TE(S,K)+TI(S,K)))*A(S,K)
C-------- PUT IN INITIAL DENSITY PROFILE FOR GAS TARGET.
      NXP1 = NX+1
      DO 5 J=1,NX
      X = NXP1-J
      Z = DX*(X-1)
      ZZZ=L1-Z
      Y = DX*(J-1)
      YYY = (L4-Y)
      XN=(.999/(1.+EXP(ZZZ/L2))+.001)*(.999/(1.+EXP(YYY/L5))+.001)
      DO 5 S=1,N
    5 N(S,J)  = XN*N(S,J)
C-------- INITIALIZE LHEAT.
      TL=0.0
      DTL=DX/CS
      WRITE(6,601)
  601 FORMAT('1',' INITIAL CONDITIONS')
      WRITE(6,602)  LO,RO,NO,TO,DO,PO,VO
  602 FORMAT('0LO=',1PE9.2,T17,'RO=',E9.2,T32,'NO=',E9.2,T47,'TO=',
     1E9.2,T62,'DO=',E9.2,T77,'PO=',E9.2,T92,'VO=',E9.2)
      WRITE(6,603)  NX,N,AO,DX,DT,PHO,THO
  603 FORMAT(' NX=',I3,T17,' N=',I3,T32,' AO=',1PE9.2
     1,T47,'DX=',E9.2,T62,'DT=',E9.2,T77,'PHO=',E9.2 ,T92,'THO=',E9.2)
      WRITE(6,604) DEL,SIGMA,EPS20,PCOOL,ABO
```

```
 604 FORMAT(' DEL=',1PE9.2,T19,'SIGMA=',E9.2,T36,'EPS20=',E9.2
    1,T53,'FCOLL=',E9.2,T70,'ABO=',E9.2)
     WRITE(6,605) DELE1,DELE2,DELI1,DELI2,LLMDA
 605 FORMAT(' DELE1=',1PE9.2,T19,'DELE2=',E9.2,T36,'DELI1=',E9.2
    1,T53,'DELI2=',E9.2,T70,'LLMDA=',E9.2)
     IF(RDN) WRITE(6,606)
 606 FORMAT('0 FULL RADIAL DYNAMICS ARE CALCULATED.')
     IF(.NOT.RDN) WRITE(6,607)
 607 FORMAT('0 RADIAL PRESSURE BALANCE IS ASSUMED.')
   2 RETURN
     END
     SUBROUTINE TPOUT(NTAPE,NTIME,TIME,M,MX,LO,RO,NO,TO,DT,A
    1,TE,TI,N,PHI,V,P,NBOUND,XLAS,U,W,XMIN,RDN,RN)
C*************************************************************************
C     OUTPUT DATA FROM SHELL ONTO DISK OR TAPE IN UNFORMATTED FORM.     *
C          RICHARD D. MILROY     77-04-25                                *
C*************************************************************************
     REAL TIME,LO,RO,NO,TO,DT,A(30,60),TE(30,60),TI(30,60),N(30,60)
    1,PHI(31),V(30,60),P(120),XLAS,U(30,60),W(30,60)
     INTEGER XMIN
     LOGICAL RDN
     WRITE(3) NTAPE,NTIME,TIME,M,MX,LO,RO,NO,TO,DT,NBOUND,XLAS
    1     ,XMIN,RDN,RN
     WRITE(3) A
     WRITE(3) TE
     WRITE(3) TI
     WRITE(3) N
     WRITE(3) PHI
     WRITE(3) V
     WRITE(3) P
     WRITE(3) U
     WRITE(3) W
     RETURN
     END
     SUBROUTINE RITE(NTIME,TIME,M,MX,A,TE,TI,N,PHI,V,P,TNO,DT,U,R,EP
    1     ,XMIN,RPB)
     REAL PHI(31),A(30,60),TE(30,60),TI(30,60),N(30,60),V(30,60)
    1,B(31),TIME,PRESS(31),P(120),TNO,DT,U(30,60),EP(30,60)
    2,R(30,60),VP(31)
     INTEGER S,X,XMIN
     LOGICAL RPB
     COMMON /PRT/ NXI,NXF,M1,MF,MS
     NNXI = NXI
     NNX = NXF
     IF(NXI.EQ.999) NNXI = XMIN
     IF(NXF.EQ.999) NNX = MX
     MMF=MF
     IF(MF.EQ.999) MMF=M
     TN=TIME*TNO
     DTS = DT*TNO
     WRITE(6,601) NTIME,TIME,TN,DT,DTS
 601 FORMAT('-NTIME=',I4,'    TIME=',1PD9.2,'    OR',D9.2,'   SECONDS'
    1,'    DT=',D9.2,'  OR',D9.2,'  SECONDS')
     WRITE(6,602)
 602 FORMAT('0    A')
     DO 1 X=NNXI,NNX
   1 WRITE(6,603) (A(S,X),S=M1,MMF,MS)
 603 FORMAT(' ',1P20D10.2)
     WRITE(6,604)
 604 FORMAT('0    TE')
```

```
      DO 2 I=NNXI,NNX
    2 WRITE(6,603) (TE(S,I),S=N1,NMP,NS)
      WRITE(6,605)
  605 FORMAT('0   TI')
      DO 3 I=NNXI,NNX
    3 WRITE(6,603) (TI(S,I),S=N1,NMP,NS)
      WRITE(6,606)
  606 FORMAT('0   N')
      DO 4 I=NNXI,NNX
    4 WRITE(6,603) (N(S,I),S=N1,NMP,NS)
      WRITE(6,607)
  607 FORMAT('0   V')
      DO 5 I=NNXI,NNX
    5 WRITE(6,603) (V(S,I),S=N1,NMP,NS)
      IF(RPB) GO TO 9
      WRITE(6,612)
  612 FORMAT('0   VP')
      DO 13 I=NNXI,NNX
      DO 14 S=1,N
   14 VP(S) = U(S,I)/N(S,I)
   13 WRITE(6,603) (VP(S),S=N1,NMP,NS)
    9 WRITE(6,608)
  608 FORMAT('0   B')
      DO 6 I=NNXI,NNX
      DO 7 S=1,N
    7 B(S)=PHI(S)/A(S,I)
    6 WRITE(6,603) (B(S),S=N1,NMP,NS)
      WRITE(6,609)
  609 FORMAT('0   R BOUNDARIES')
      DO 8 I=NNXI,NNX
    8 WRITE(6,603) (R(S,I),S=N1,NMP,NS)
      WRITE(6,610)
  610 FORMAT('0   PRESSURE')
      DO 10 I=NNXI,NNX
      DO 11 S=1,N
   11 PRESS(S)=N(S,I)*(TE(S,I)+TI(S,I))+(PHI(S)/A(S,I))
   10 WRITE(6,603) (PRESS(S),S=N1,NMP,NS)
      WRITE(6,614)
  614 FORMAT('0   EP')
      DO 15 I=NNXI,NNX
   15 WRITE(6,603) (EP(S,I),S=N1,NMP,NS)
      WRITE(6,611)
  611 FORMAT('0   LASER POWER')
      NXMP1=NX-NNXI+1
      DO 12 I=1,NXMP1
   12 WRITE(6,603) P(I),P(I+NX)
      RETURN
      END
      SUBROUTINE REST(DT,DX,DEL,PCOLL,EPS20,PHI,V,N,A,TE,TI
     1,DELE1,DELE2,DELI1,DELI2,ABO,CS,P,P2,TIME,THO,XLAS,R,U,NO
     2,NMIN)
C*****************************************************************
C     *  READS OFF FORTRAN UNIT 02 TO RESTART PROGRAM AT RECORD NT.  *
C     *        RICHARD B. MILROY          JUNE 1977.                  *
C*****************************************************************
      IMPLICIT REAL  (A-H,O-Z)
      REAL  A(30,60),TE(30,60),TI(30,60),N(30,60),PHI(31),V(30,60)
     *,P(120),LMDA,LLSDA,LO,MO,MI,KB,P2(31),PR(30,60),R(30,60),U(30,60)
     *,NO(30,60),L1,L2,L3,NCBIT
      INTEGER NMIN
```

```
      LOGICAL TRBC,RDN
      COMMON L0,R0,N0,T0,PWR1,PWR2,PWR3,T1,T2,T3,T4,TMAX,VLAS,TDP1,TDP2
     *,RM
     1,TS,AVR,AVX,PP,NX,N,NTOTAL,NBOUND,NTIME,NPRT,NTAPE,NXMIN,RDN,TRBC
     2/TSS/A11,A1,A2,A3,DTMAX/LASR/L3,RL0,BS,NCRIT,TL,DTL,NRMAX
     3/PRT/ NXI,NXF,N1,NF,NS
      NAMELIST /INPUT/ NTIME,NTOTAL,NPRT,NTAPE,PR,PWR1,PWR2,PWR3
     1,T1,T2,T3,T4,TMAX,NBOUND,NX,N,L0,R0,N0,T0,R01,U,VLAS,RM
     2,BINT,TINT,PP,AVR,A11,A1,A2,A3,L1,L2,L3,RL0,BS,DTMAX
     3,NXMIN,NXMIN1,RDN,TDP1,TDP2,NXI,NXF,N1,NF,NS,AVX,NRMAX,TRBC
    1 READ(2) NTAPE,NT,TIME,N,NX,L0,R0,N0,T0,DT,NBOUND,XLAS
     1      ,XMIN,RDN,RM
      READ(2) A
      READ(2) TE
      READ(2) TI
      READ(2) N
      READ(2) PHI
      READ(2) V
      READ(2) P
      READ(2) U
      READ(2) NO
      IF(NT.NE.NTIME) GO TO 1
      READ(5,INPUT)
      DO 2 J=1,NX
      DO 2 I=1,N
      R(I,J) = SQRT(NO(I,J))
    2 CONTINUE
      NX2=2*NX
      DO 4 I=1,NX2
    4 P2(I)=P(I)
      L3=L3/L0
      NCRIT = 9.96D+18/N0
      RL0=RL0/R0
      BS=BS*L0/R0
      DX=1.0/(NX-NBOUND)
      KB=1.6D-12
      C=3.0D+10
      MI=1.67D-24*RM
      P0=N0*KB*T0
      B0=SQRT(8.0*3.1415*P0)
      V0=SQRT(KB*T0/MI)
      TH0=L0/V0
      A0=3.14159*R0**2
      PW0=P0*A0*V0*1.0E-07
      CS=C/V0
      LMDA=1.356E+28*((KB*T0)**3/(3.14159*N0))**0.5
      LLMDA=ALOG(LMDA)
      DEL=1.9D21*T0**2.5/(LLMDA*N0*V0*L0)
      DELE1 = (1.9D+21/LLMDA)*(T0**2.5*L0)/(N0*V0*R0**2)
      DELE2 = (2.45D+25/LLMDA**2)*(T0**3*B0**2)/(N0**2)
      DELI1 = (7.8D+19/LLMDA)*(T0**2.5*L0)/(N0*V0*R0**2)*SQRT(RM)
      DELI2 = (7.8D+22/LLMDA**2)*(T0**3*B0**2)/(N0**2)
      SIGMA=8.7D13*T0**1.5/LLMDA
      EPS20=C**2*L0/(6.28319*SIGMA*R0**2*V0)
      PCOLL=3.25D-09*N0*LLMDA/(T0**1.5)*L0/(V0*RM)
      AB0=9.74E-36*N0**2*L0/T0**1.5
      PWR1 = PWR1/PW0
      PWR2 = PWR2/PW0
      PWR3 = PWR3/PW0
      T1 = T1/TH0
```

```
      T2 = T2/TN0
      T3 = T3/TN0
      T4 = T4/TN0
      T5 = T4+3.0/CS
      TMAX = TMAX/TN0
      VLAS=VLAS/V0
      DTMAX = DTMAX/TN0
C------- INITIALIZE LHEAT.
      TL = TIME
      DTL = DX/CS
    5 WRITE(6,601)
  601 FORMAT('1',' INITIAL CONDITIONS')
      WRITE(6,602) L0,R0,N0,T0,B0,P0,V0
  602 FORMAT(' L0=',1PE9.2,T17,'R0=',E9.2,T32,'N0=',E9.2,T47,'T0=',
     1E9.2,T62,'B0=',E9.2,T77,'P0=',E9.2,T92,'V0=',E9.2)
      WRITE(6,603) NX,M,A0,DX,DT,PH0,TN0
  603 FORMAT(' NX=',I3,T17,'M=',I3,T32,'A0=',1PE9.2
     1,T47,'DX=',E9.2,T62,'DT=',E9.2,T77,'PH0=',E9.2 ,T92,'TN0=',E9.2)
      WRITE(6,604) DEL,SIGMA,EPS20,PCOLL,AB0
  604 FORMAT(' DEL=',1PE9.2,T19,'SIGMA=',E9.2,T36,'EPS20=',E9.2
     1,T53,'PCOLL=',E9.2,T70,'AB0=',E9.2)
      WRITE(6,605) DELE1,DELE2,DELI1,DELI2,LLMDA
  605 FORMAT(' DELE1=',1PE9.2,T19,'DELE2=',E9.2,T36,'DELI1=',E9.2
     1,T53,'DELI2=',E9.2,T70,'LLMDA=',E9.2)
      IF(RDN) WRITE(6,606)
  606 FORMAT('0 FULL RADIAL DYNAMICS ARE CALCULATED.')
      IF(.NOT.RDN) WRITE(6,607)
  607 FORMAT('0 RADIAL PRESSURE BALANCE IS ASSUMED.')
      RETURN
      END
      SUBROUTINE LHEAT(M,NX,STEP,TIME,T1,T2,T3,T4,NBOUND,TE15,XMIN
     1,PWR1,PWR2,PWR3,XLAS,VLAS,R0,AB0,P,P2,PR,R,N,A,TE,EP,DX,NXP1,T0)
C*********************************************************************
C     *                  FIND LASER POWER.                           *
C     *          P IS POWER IN UNITS OF (N0*KB*T0*A0*V0)             *
C     *   E=(UNITS OF POWER)/(UNIT OF LENGTH) DEPOSITED IN PLASMA.   *
C     *            RICHARD D. MILROY   MAY 15, 1977.                 *
C*********************************************************************
      IMPLICIT REAL(A-H,O-Z)
      REAL P(120),PR(30,60),R(30,60),N(30,60),A(30,60),TE(30,60)
     1,EP(30,60),PSI(120),P2(120),E(120),KA(120),LMDA,LLMDA(30,60),L3
     2,NCRIT,NCOR,TE15(30,60),KA1(120)
      REAL TDIF
      INTEGER S,X,STEP,XMIN
      COMMON/LASP/L3,ZL0,BS,NCRIT,TL,DTL,NRMAX
      T015 = T0*SQRT(T0)
C------- CALCULATE NR = NO. OF LASER TIMESTEPS TO TAKE.
      NXP2 = NX + 2
      NX2 = 2*NX
      NX2M2 = NX2 - 2
      TDIF = TIME - TL
      NR = (TDIF/DTL + 0.5)
      IF (NR.LT.0) STOP
      NR=MIN0(NR,NRMAX)
      IF(STEP.EQ.1) NR=NR/2
      IF (NR.EQ.0) RETURN
      DT = 0.0D0
      IF (STEP.EQ.2) DT=NR*DTL
      TL = TL+DT
C------- LASER PROP. DIST. LIMITED BY BLEACHING WAVE VELOCITY - VLAS.
```

```
      XLAS=XLAS+VLAS*DT
      NLAS=(XLAS/DX) + 2
      NXP2NX = NX+2-XMIN
      NLAS = MIN0(NXP1,NLAS,NXP2NX)
C-------- DEFINE PR(S,X).  RELATIVE RADIAL INTENSITY.
C-------- ASSUME GAUSSIAN PROFILE.
      DO 6 X=XMIN,NX
      J = NXP1 - X
      Z = DX*(J-1)
      SIGMA = RLO + BS*ABS(L3-Z)
      SIG2 = 2.0*SIGMA**2
      RS = 0.0
      DO 6 S=1,N
      RS = 0.5*(RS+R(S,X))
      EXZZ = RS**2/SIG2
      IF(EXZZ.GT.100.0) EXZZ = 100.0D0
      PR(S,X) = EXP(-EXZZ)
      RS = R(S,X)
    6 CONTINUE
C--------- DEFINE LASER POWER AT FIRST X - POINT.
      IF (TIME.GT. T1) GO TO 15
      P1 = PWR1*TIME/T1
      GO TO 85
   15 CONTINUE
      IF (TIME.GT.T2) GO TO 16
      P1 = PWR1+(PWR2-PWR1)*(TIME-T1)/(T2-T1)
      GO TO 85
   16 CONTINUE
      IF (TIME.GT.T3) GO TO 17
      P1 = PWR2+(PWR3-PWR2)*(TIME-T2)/(T3-T2)
      GO TO 85
   17 CONTINUE
      IF (TIME.GT.T4) GO TO 18
      P1 = PWR3-PWR3*(TIME-T3)/(T4-T3)
      GO TO 85
   18 P1 = 0.0D00
   85 CONTINUE
C---------- DEFINE PSI(X)
      DO 13 X=XMIN,NX
      PSI(X)=0.0
      DO 13 S=1,N
      LNDA=18.7*TE(S,X)*T0
      IF(TE(S,X)*T0.LT.27.0) LNDA=2.3*(T015*TE15(S,X))
      IF(N(S,X).GT.(0.9999*NCRIT)) GO TO 20
      NCOR = 1.0/SQRT(1.0-N(S,X)/NCRIT)
      GO TO 21
   20 NCOR=100.0
   21 CONTINUE
      LLNDA(S,X)=ALOG(LNDA)*NCOR
   13 PSI(X)=PR(S,X)*A(S,X)+PSI(X)
C---------- DEFINE KA(X) - LASER ABSORPTION RATE AT X-POINTS.
      DO 1 X=XMIN,NX
      KA(X)=0.0
      DO 14 S=1,N
   14 KA(X)=KA(X)+N(S,X)**2*PR(S,X)*A(S,X)*LLNDA(S,X)/TE15(S,X)
      KA(X)=KA(X)*ABO/PSI(X)
      KA1(X) = KA(X)
      IF(KA(X).GT.(2.0/DX)) KA(X)=2.0/DX
    1 CONTINUE
C-------- INITIALIZE E(X) TO 0.0.
```

```
      DO 7 X=1,NM2
    7 E(X)=0.0
C-------- FIND LASER POWER.
      DO 2 IT=1,NR
      P3=P1
      DO 3 X=2,NLAS
      P0=P(X-1)*(1.0-0.5*KA(NX+2-X)*DX)/(1.0+0.5*KA(NX+2-X)*DX)
      E(X-1)=E(X-1)+(P(X-1)-P0)/DX
      P(X-1)=P3
    3 P3=P0
      P(NLAS)=P3
      IF(NBOUND.EQ.1) GO TO 2
      IF(NLAS.LT.NX) GO TO 2
      DO 4 X=NXP1,NX2M2
      P0=P(X-1)*(1.0-0.5*KA(X-NX+2)*DX)/(1.0+0.5*KA(X-NX+2)*DX)
      E(X-1)=E(X-1)+(P(X-1)-P0)/DX
      P(X-1)=P3
    4 P3=P0
      P(NX2M2)=P3
    2 CONTINUE
      I=NLAS
      IF(NBOUND.EQ.2) I=NX2M2
      IF(STEP.EQ.1) GO TO 10
      DO 23 X=1,I
   23 P2(X)=P(X)
      GO TO 26
   10 CONTINUE
      DO 22 X=1,I
   22 P(X)=P2(X)
   26 CONTINUE
      DO 5 X=XMIN,NX
      DO 5 S=1,M
      EP1=ABO/(KA1(X)*PSI(X))*N(S,X)*PR(S,X)*LLMDA(S,X)/TE15(S,X)
      EP(S,X) = EP1*E(NX+1-X)/NR
      IF(NBOUND.EQ.2) EP(S,X)=EP(S,X)+EP1*E(NX-3+X)/NR
    5 CONTINUE
      RETURN
      END
      SUBROUTINE DTIME(DT,NX,M,TE,TI,TE2,N,PHI,A,R,EPS1,XMIN,RDN,DX)
C**********************************************************************
C     *     CALCULATE MAXIMUM ALLOWABLE TIMESTEP 2-D ROUTINE CAN MAKE.   *
C     *     STABILITY CONDITION IS BASED ON RADIAL DYNAMICS.              *
C     *          RICHARD D. MILROY        DECEMBER 15, 1977.              *
C**********************************************************************
      IMPLICIT REAL (A-H,O-Z)
      REAL TE(30,60),TI(30,60),TE2(30,60),N(30,60),R(30,60)
     1,RINT(31),PHI(31),A(30,60)
      INTEGER XMIN,S,X
      LOGICAL RDN
      COMMON/TSS/A11,A1,A2,A3,DTMAX
      DTO = DT
      MM1 = 2
      MM2 = 1
      MXS = 1
      DTST = 1.0D+50
      DTTEM = 1.0D+50
      DZ = DX
C------- MAKE DT SATISFY RADIAL STABILITY CONDITION.
      DO 1 X=XMIN,NX,MXS
      S=1
```

```
      BS = PHI(S)/A(S,X)
      C = (1.6667*(TE(S,X)+TI(S,X))) + (2.0*BS**2/N(S,X))
      C = SQRT(C)
      IF(RDN) DZ=R(S,X)*EPS1
      DTN = A1*DZ/C
      IF(DTN.LT.DTST) DTST = DTN
      IF(MM1.LT.2) GO TO 1
      DO 2 S=2,MM1
      BS = PHI(S)/A(S,X)
      C = (1.6667*(TE(S,X)+TI(S,X))) + (2.0*BS**2/N(S,X))
      C = SQRT(C)
      IF(RDN) DZ=(R(S,X)-R(S-1,X))*EPS1
      DTN = A1*DZ/C
      IF(DTN.LT.DTST) DTST = DTN
    2 CONTINUE
    1 CONTINUE
C------- LIMIT DT SO ELECTRON TEMP. DOES NOT CHANGE TOO FAST.
      DO 3 X=XMIN,NX,NXS
      DO 3 S= 1 ,MM2
      DTE = ABS(TE(S,X) - TE2(S,X))
      IF(DTE.LT.1.0D-20) DTE = 1.0D-20
      DTN = A2*TE2(S,X)*DTO/DTE
      IF(DTN.LT.DTTEM) DTTEM = DTN
    3 CONTINUE
C------- PREVENT DT FROM INCREASING BY MORE THAN A FACTOR OF A3.
      DTN = A3*DTO
C------- SET DT TO MOST RESTRICTIVE CASE.
      DT = AMIN1(DTST,DTTEM,DTN)
      IF(DT.GT.DTMAX) DT=DTMAX
      RETURN
      END
      SUBROUTINE MDPTS(XMIN,P,NX,NBOUND,NXMIN)
C*********************************************************************
C    *     CALCULATE XMIN.                                           *
C    *     PREVENTS PROGRAM FROM DOING CALCULATIONS IN SECTIONS      *
C    *          WHERE THERE IS NO PLASMA MOTION.                     *
C    *     TO REDUCE CPU COSTS.                                      *
C    *           RICHARD D. MILROY.     78-01-06                     *
C*********************************************************************
      REAL P(120)
      INTEGER XMIN
      NXP1 = NX+2-NBOUND-NXMIN
      N1 = NX-XMIN+3-NXMIN
      K=N1-1
      DO 1 J=N1,NXP1
      IF(P(J).LT.1.0D-10) GO TO 2
    1 K=J
    2 NX2 = NX-K+2
      NX2 = NX2-NXMIN
      IF(XMIN.GT.NX2) XMIN=NX2
      RETURN
      END
```

```
C*****************************************************************************
C    *    TEST FOR CONSERVATION OF ENERGY IN 2-D M.H.D. ROUTINE.             *
C    *    MARCH, 27, 1977.                                                   *
C    *    MODIFIED APRIL, 30, 1977.                                          *
C    *    MODIFIED AGAIN (TO INCLUDE CHECK ON CONSERVATION OF MASS)          *
C    *    ON JANUARY, 18, 1978.                                              *
C    *     WRITTEN AND MODIFIED BY RICHARD D. MILROY.                        *
C*****************************************************************************
      REAL A(30,60),TE(30,60),TI(30,60),N(30,60),PHI(31)
     1,V(30,60),LO,NO,LLMDA,P(120),W(30,60),VP(30,60)
      LOGICAL RDN
      COMMON A,TE,TI,N,PHI,V,LO,NO,VO,T,NS,NZ,DZ,DEL,GM1I,GG,P,W,VP
      COMMON/CONST/ NBOUND,NBR,NBL,N1,N2,N3,N4,N5,N6,N7,N8,EPS1
C---------READ I. WILL CALCULATE ENERGY OF EVERY I'TH RECORD.
      WRITE(6,601)
  601 FORMAT('-','ENTER I - I3 FORMAT. - AND TP'/' ','  I          TP')
      READ(5,501) IX,TP
  501 FORMAT(I3,E10.3)
      GAM=5.0/3.0
      GM1I=1.0/(GAM-1.0)
      GG=GAM*GM1I
C--------CALCULATE DEL FOR THERMAL CONDUCTIVITY OUT END.
      READ(3) NTAPE,NTIME,T,NS,NZ,LO,RO,NO,TO,DT,NBOUND,XLAS
     1,XMIN,RDN,RM
      TMO = 1.02E-06*LO/SQRT(TO)*SQRT(RM)
      NBL=2
      NBR=2
      IF(NBOUND.EQ.2) NBL=1
      N1=NBL+1
      N2=NBL+2
      N3=NZ-NBR-1
      N4=NZ-NBR
      N5=NBR+2
      N6=NZ-NBL
      N7=2*NZ-3-NBR
      N8=N4-N1
      BACKSPACE 3
      EPS1 = RO/LO
      DU1 = 0.0
      DG1 = 0.0
      LLMDA=ALOG(1.356E+28*((1.6E-12*TO)**3/(3.14159*NO))**0.5)
      VO=SQRT(1.6E-12*TO/(1.67E-24*RM))
      DEL=1.9E21*TO**2.5/(LLMDA*NO*VO*LO)
      IT=1
      IF(IX.NE.1) IT=2
C---------CALCULATE ENERGY ETC. IN 1'ST RECORD.
      CALL RD(DU,DG,&4)
      T2=T
      CALL ENG(U1,G1)
      DUT=DU
      DGT=DG
      TIME = T*TMO
      WRITE(6,602) TIME,U1,G1
  602 FORMAT(' ','AT ',1PE10.3,' SEC.  U= ',E10.3,'    MASS = ',E10.3)
      GO TO(3,2),IT
    2 CONTINUE
C---------FIND DU AT INTERMEDIATE TIMESTEPS.
      IM1=IX-1
      DO 1 I=1,IM1
```

```
      CALL RD(DU,DG,&4)
      T1=T
      DELTT=T1-T2
      T2=T1
      DU1=DU1+0.5*DELTT*(DUT+DU)
      DG1=DG1+0.5*DELTT*(DGT+DG)
      DUT=DU
    1 DGT=DG
    3 CONTINUE
C---------FIND ENERGY ETC. IN I'TH RECORD.
      CALL RD(DU,DG,&4)
      T1=T
      DELTT=T1-T2
      T2=T1
      DU1=DU1+0.5*DELTT*(DUT+DU)
      DG1=DG1+0.5*DELTT*(DGT+DG)
      DUT=DU
      DGT=DG
      CALL ENG(U2,G2)
      DIFF=U2-U1
      DIFG=G2-G1
      FRACT=(DIFF+DU1)*2.0/(U1+U2)
      FRACG=(DIFG+DG1)*2.0/(G1+G2)
      WRITE(6,603) DIFF,DU1,FRACT
  603 FORMAT(' ','DIFF= ',1PE10.3,'   ENERGY LOST OUT END = ',E10.3,
     1'   FRACTIONAL ERROR=',E10.3)
      WRITE(6,604) DIFG,DG1,FRACG
  604 FORMAT(' ','DIFF= ',1PE10.3,'   MASS LOST OUT END =   ',E10.3,
     1'   FRACTIONAL ERROR=',E10.3)
      TIME = T*TNO
      WRITE(6,602) TIME,U2,G2
      U1=U2
      G1=G2
      DU1=0.0
      DG1=0.0
      IF(T.GT.TF) IT=3
      GO TO(3,2,4),IT
    4 STOP
      END
      SUBROUTINE RD(DU,DG,*)
C---------READS DATA & CALCULATES ENERGY & MASS LOST FROM ENDS.
      REAL A(30,60),TE(30,60),TI(30,60),N(30,60),PHI(31)
     1,V(30,60),LO,NO,P(120),W(30,60),VP(30,60)
      COMMON A,TE,TI,N,PHI,V,LO,NO,VO,T,NS,NZ,DZ,DZL,GM1I,GG,P,W,VP
      COMMON/CONST/ NBOUND,NBR,NBL,N1,N2,N3,N4,N5,N6,N7,N8,EPS1
      READ(3,END=1) NTAPE,NTIME,T,NS,NZ,LO,RO,NU,TO,DT,NBOUND
      DZ=1.0/(NZ-NBOUND)
      READ(3) A
      READ(3) TE
      READ(3) TI
      READ(3) N
      READ(3) PHI
      READ(3) V
      READ(3) P
      READ(3) VP
      READ(3) W
      DO 7 J=1,NZ
      DO 7 I=1,NS
    7 VP(I,J) = VP(I,J)/SQRT(N(I,J))
      DO 5 J=1,NZ
```

```
      VPS1 = 0.5*VP(1,J)
      DO 6 I=2,NS
      VPS2 = 0.5*(VP(I-1,J) + VP(I,J))
      VP(I-1,J) = VPS1
    6 VPS1 = VPS2
    5 VP(NS,J)=VPS1
      DU=0.0
      DO 2 I=1,NS
      DU=DU+(A(I,N4)*(GG*N(I,N4)*(TE(I,N4)+TI(I,N4))+0.5*N(I,N4)*
     1(V(I,N4)**2+(EPS1*VP(I,N4))**2))*V(I,N4)*DT)
     2+DEL*A(I,N4)*TE(I,N4)**2.5*0.5*(TE(I,N4-1)-TE(I,N4+1))*(DT/DZ)
    2 CONTINUE
      IF(NBOUND.EQ.2) GO TO 4
      DO 3 I=1,NS
      DU=DU-(A(I,N1)*(GG*N(I,N1)*(TE(I,N1)+TI(I,N1))+0.5*N(I,N1)*
     1(V(I,N1)**2+(EPS1*VP(I,N1))**2)*V(I,N1)*DT)
     2-DEL*A(I,N1)*TE(I,N1)**2.5*0.5*(TE(I,N1-1)-TE(I,N1+1))*(DT/DZ)
    3 CONTINUE
C-------- ADD LASER ENERGY INPUT.
    4 CONTINUE
      IF(NBOUND.EQ.1) DU=DU-0.5*(P(NBR+1)+P(NBR+2)-P(N6)-P(N6+1))*DT
      IF(NBOUND.EQ.2) DU=DU-0.5*(P(NBR+1)+P(N3R+2)-P(N7)-P(N7+1))*DT
C------- CALCULATE MASS LOST FROM SOLENOID ENDS.
      DG = 0.0
      DO 8 I=1,NS
    8 DG = DG + A(I,N4)*V(I,N4)*N(I,N4)*DT
      IF(NBOUND.EQ.2) GO TO 9
      DO 10 I=1,NS
   10 DG = DG - A(I,N1)*V(I,N1)*N(I,N1)*DT
    9 CONTINUE
      DU=DU/DT
      DG=DG/DT
      RETURN
    1 WRITE(6,601)
  601 FORMAT(' ','ENDFILE ON UNIT #3')
      RETURN1
      END
      SUBROUTINE ENG(U,G)
C---------CALCULATE TOTAL ENERGY & MASS IN SOLENOID.
      REAL A(30,60),TE(30,60),TI(30,60),N(30,60),PHI(31)
     1,V(30,60),LO,NO,P(120),W(30,60),VP(30,60)
      REAL*8 UD,DBLE,GD
      COMMON A,TE,TI,N,PHI,V,LO,NO,VO,T,NS,NZ,DZ,DEL,GM1I,GG,P,W,VP
      COMMON/CONST/ NBOUND,NBR,NBL,N1,N2,N3,N4,N5,N6,N7,N8,EPS1
      UD=0.0D0
      DO 1 J=N2,N3
      DO 1 I=1,NS
      B=PHI(I)/A(I,J)
      U=A(I,J)*(GM1I*N(I,J)*(TE(I,J)+TI(I,J))+B**2+0.5*N(I,J)
     1*(V(I,J)**2+(EPS1*VP(I,J))**2))
      UD=UD+DBLE(U)
    1 CONTINUE
      DO 2 J=N1,N4,N8
      DO 2 I=1,NS
      B=PHI(I)/A(I,J)
      U=A(I,J)*(GM1I*N(I,J)*(TE(I,J)+TI(I,J))+B**2+0.5*N(I,J)
     1*(V(I,J)**2+(EPS1*VP(I,J))**2))
      UD=UD+0.5D0*DBLE(U)
    2 CONTINUE
C-------- ADD ENERGY OF LASER BEAM.
```

```
      PENG=0.0D0
      DO 3 J=N5,N6
    3 PENG=PENG+P(J)
      IF(NBOUND.EQ.1) GO TO 5
      DO 4 J=N2,N7
    4 PENG=PENG+P(J)
    5 CS=3.0D+10/V0
      PENG=PENG/CS
      UD=UD+PENG
C--------- CALCULATE FINAL VALUE OF ENERGY.
      U=SNGL(UD)
      U=U*DZ
C-------- CALCULATE TOTAL MASS IN SOLENOID.
      GD = 0.0
      DO 6 J=N2,N3
      DO 6 I=1,NS
      G = A(I,J)*N(I,J)
    6 GD = GD+DBLE(G)
      DO 7 J=N1,N4,N8
      DO 7 I=1,NS
      G = A(I,J)*N(I,J)
    7 GD = GD+0.5*DBLE(G)
      G=SNGL(GD)
      G=G*DZ
      RETURN
      END
```

```
C*****************************************************************************
C    *    DRAW 3-D PLOTS FOR 2-D MHD ROUTINE.                                *
C    *   RICHARD D. MILROY     1977                                          *
C*****************************************************************************
      REAL A(30,60),TE(30,60),TI(30,60),N(30,60),B(30,60)
     1,V(30,60),R(30,60),PHI(30),Z(62),RX(62)
     2,D(100,100),RT(30,60),RX1(30,60)
     3,P(120),PT(30),LO,NO,W(30,60),VPS(30,60),NMAX
      LOGICAL RDN
      COMMON/AREA1/DIST,PITCH,SIZE,KODE,NGN,NIR,NIQ,NIL
      NNZ = 60
      NR=30
      NIR=4
      NIL=4
      NIQ=5
      NMAX = 1.999E+19
      DIST = 5.0
      YAW = 45.0
      PITCH = 30.0
      SIZE = 5.0
      KODE = 0
      MGN = 2
      WRITE(6,601)
  601 FORMAT(' INPUT NO OF PLOTSETS - I3 FORMAT')
      READ(5,501) NPLTS
  501 FORMAT(I3)
      WRITE(6,602)
  602 FORMAT(' INPUT DESIRED PLOT TIMES')
      READ(5,502) (PT(I),I=1,NPLTS)
  502 FORMAT(10E10.3)
      READ(3) NTAPE,NTIME,T,NS,NZ,LO,RO,NO,TO,DT,NBOUND,XLAS
     1,XMIN,RDN,RM
      TNO=LO/SQRT(9.58E+[illegible]) *SQRT[illegible])
      DO 1 I=1,NPLTS
    1 PT(I)=PT(I)/TNO
      NS2=NS+2
      NZ2=NZ+2
      NR2=NR+2
      CALL PLOTS
      CALL ORG1(SIZE,SIZE)
      CALL NPLT2(N[illegible],NZ,NNZ,N[illegible],NS2,NZ2,[illegible],N[illegible]TS,SIZE,NMAX
     1,A,TE,TI,N,B,[illegible]R,PHI,Z,RX,D,RT,RX1
     2,N[illegible],W,VPS,YAW,RM,RDN)
      CALL PLOT(0.0,0.0,999)
      STOP
      END
      SUBROUTINE NPLT2(NS,NZ,NNZ,NR
     1,NS2,[illegible],PT,NPLTS,SIZE,NMAX
     2,A,TE,[illegible],N,B,V,R,PHI,Z,RX,D,RT,RX1,NR2
     3,W,VPS,YAW,RM,RDN)
      REAL A(30,60),TE(30,60),TI(30,60),N(30,60),B(30,60)
     1,V(30,60),R(30,60),PHI(30),Z(NZ2),RX(NZ2)
     2,RT(30,60),LO,NO,D(NR,NNZ),RX1(30,60)
     3,P(120),PT(30),W(30,60),VPS(30,60),NMAX
      LOGICAL RDN
      BACKSPACE 3
      N[illegible]
      T=1.0E50
    1 TOLD=T
```

```
      READ(3,END=23) NSTEP,NTIME,T,NS,NZ,LO,RO,PO,TO,DR,ENOUGH
      IF(T.LE.1.0E-30) T=1.0E-30
      IF(T.GT.PT(NP)) GO TO 20
      READ(3) A
      READ(3) TS
      READ(3) TI
      READ(3) E
      READ(3) PHI
      READ(3) V
      READ(3) P
      READ(3) VPS
      READ(3) U
      GO TO 1
   20 CONTINUE
      IF(ABS(PT(NP)-T).GT.ABS(PT(NP)-TOLD)) GO TO 21
      NP=NP+1
      READ(3) A
      READ(3) TS
      READ(3) TI
      READ(3) E
      READ(3) PHI
      READ(3) V
      READ(3) P
      READ(3) VPS
      READ(3) U
      GO TO 22
   21 BACKSPACE 3
      T=TOLD
      NP=NP+1
      GO TO 22
   23 NP=NPLTS+1
   22 CONTINUE
C-------- DEFINE B(S,Z).
      DO 11 J=1,NZ
      DO 11 I=1,NS
   11 B(I,J)=PHI(I)/A(I,J)
C---------- FIND R(S,Z)
      DO 4 J=1,NZ
      JJ = NZ+1-J
      AT=0.0
      ATO=0.0
      DO 5 I=1,NS
      AT=AT+A(I,J)
      R(I,JJ)=0.5*(SQRT(ATO)+SQRT(AT))
      RI1(I,JJ)=SQRT(AT)
      ATO=AT
    5 CONTINUE
    4 CONTINUE
C------- PLOT CURRENT TIME IN SECONDS.
      TIME=T*LO/SQRT(9.58E+11*TO) *SQRT(RM)
      NEXP=ALOG10(TIME)
      NEXP=NEXP-1
      EXP=NEXP
      BASE=TIME/(10.0**NEXP)
      HT=(SIZE-2.0)/15.0
      CALL SYMBOL(1.0,1.0,HT,'TIME=      X 10',0.0,14)
      VX=1.0+5.0*HT
      CALL NUMBER(VX,1.0,HT,BASE,0.0,2)
      VX=1.0+14.0*HT
      VY=1.0+0.5*HT
```

```
      NT=0.5*NT
      CALL NUMBER(VX,VY,HT,EXP,0.0,-1)
      CALL ORG(SIZE,SIZE)
C-------- CONVERT QUANTITIES TO MORE NATURAL UNITS.
C-- THE TI IN EV.  V IN 10**7 CM/SEC.  B IN KG.  N IN 10**17 CM**-3.
      VO = 9.788E+05*SQRT(TO)/SQRT(AM)
      BO = 6.341E-09*SQRT(NO*TO)
      DO 12 J=1,NZ
      DO 12 I=1,NS
      TE(I,J) = TO*TE(I,J)
      TI(I,J) = TO*TI(I,J)
      V(I,J) = VO*V(I,J)
      VPS(I,J) = VO*RO/LO*VPS(I,J)/RX1(I,J)
      B(I,J) = BO*B(I,J)
      N(I,J) = NO*N(I,J)
      IF(N(I,J).GT.NMAX) N(I,J) = NMAX
   12 CONTINUE
      DO 13 J=1,NZ
   13 VPS(1,J)=0.0
C-------- PLOT VARIOUS QUANTITIES
      CALL BDND(SIZE,RX1,RX,Z,NS,NZ,NZ2,LO,RO)
      CALL PL3DS(NS,NZ,NR,NNZ,NBOUND,TE,D,R,YAW,RO,LO,'TE',2)
      CALL PL3DS(NS,NZ,NR,NNZ,NBOUND,TI,D,R,YAW,RO,LO,'TI',2)
      CALL PL3DS(NS,NZ,NR,NNZ,NBOUND,V,D,R,YAW,RO,LO,'VZ',2)
      IF (BDN)
     *CALL PL3DS(NS,NZ,NR,NNZ,NBOUND,VPS,D,R,YAW,RO,LO,'VR',2)
      YAW = 360.0-YAW
      CALL PL3DS(NS,NZ,NR,NNZ,NBOUND,B,D,R,YAW,RO,LO,'B',1)
      CALL PL3DS(NS,NZ,NR,NNZ,NBOUND,N,D,R,YAW,RO,LO,'N',1)
      YAW = 360.0-YAW
      CALL LASPLT(F,NZ,NO,TO,RO,LO,NBOUND)
      IF(NP.LE.NPLTS) GO TO 1
  999 RETURN
      END
      SUBROUTINE RORG(Q,QN,R,NS,NZ,NR,NNZ,NBOUND)
C------- SUBROUTINE RE-ORGANIZES DATA TO RECTANGULAR GRID QN.
C------- LINEAR INTERPOLATION IS USED.
      REAL Q(30,60),R(30,60),QN(NR,NNZ)
C------- REVERSE ORDER OF 2'ND INDEX IN ARRAY Q.
      N = NZ/2
      DO 1 J=1,N
      N1 = NZ+1-J
      DO 1 I=1,NS
      TEMP = Q(I,J)
      Q(I,J) = Q(I,N1)
    1 Q(I,N1) = TEMP
C------- FIND QN.
      DZN = 1.0/(NNZ-1)
      DZO = 1.0/(NZ-NBOUND)
      DO 2 J=1,NNZ
      Z = (J-1)*DZN
      L = Z/DZO + 1
      IF(L.EQ.NZ) GO TO 7
      LP1 = L+1
      Z1 = (L-1)*DZO
      Z2 = L*DZO
      A2 = (Z-Z1)/(Z2-Z1)
      A1 = 1.0 - A2
      GO TO 8
    7 L = NZ-1
```

```
      LP1 = NZ
      A1 = 0.0
      A2 = 1.0
      DRN = 1.0/(NR-1)
      K = 1
      DO 2 I=1,NR
    3 CONTINUE
      RN = (I-1)*DRN
      RO = A1*R(K,L) + A2*R(K,LP1)
      IF(RO.GT.RN) GO TO 4
      IF(K.GE.NS) GO TO 5
      K = K+1
      GO TO 3
    4 KM1 = K-1
      IF(K.EQ.1) GO TO 5
      RO1 = A1*R(KM1,L) + A2*R(KM1,LP1)
      B = (RN-RO1)/(RO-RO1)
      QM = A1*Q(KM1,L) + A2*Q(KM1,LP1)
      QP = A1*Q(K,L) + A2*Q(K,LP1)
      QN(I,J) = QM + B*(QP-QM)
      GO TO 2
    5 QN(I,J) = A1*Q(K,L) + A2*Q(K,LP1)
    2 CONTINUE
      RETURN
      END
      SUBROUTINE ORG1(XLEN,YLEN)
C---------MOVE ORIGIN IN A WAY TO MINIMIZE PLOTTER PAPER WASTE.
C---------FOR TEKTRONIX, RE-SCALE PLOT AND HALT UNTILL RETURN ENTERED.
      CALL TCLEAR(61)
      CALL TERASE(61)
      F=AMIN1(30.0/XLEN,20.0/YLEN)
      CALL FACTOR(F)
      CALL PLOT(2.0,2.0,-3)
      GO TO 10
    1 CALL PLOT(3.0,3.0,-3)
      X0=0.0
      Y0=0.0
      XL=XLEN
      YL=YLEN
      GO TO 10
      ENTRY ORG(XLEN,YLEN)
C---------ENTER HERE FOR ALL BUT 1'ST CALL TO THIS ROUTINE.
      CALL TCLEAR(62)
      READ(6,601) NOTHG
  601 FORMAT(I5)
      CALL TERASE(62)
      READ(6,601) NOTHG
      GO TO 10
    2 VX=6.0+XL
      VY=6.0+YL
      Y=Y0+VY+YLEN
      YL=YLEN
      IF(Y.GT.33.0) GO TO 3
      CALL PLOT(0.0,VY,-3)
      Y0=Y0+VY
      IF(XLEN.GT.XL) XL=XLEN
      GO TO 10
    3 VY=-Y0
      CALL PLOT(VX,VY,-3)
      XL=XLEN
```

```
      X0=X0+VX
      Y0=0.0
   10 RETURN
      END
      SUBROUTINE PL3DS(NS,NZ,NR,NNZ,NBOUND,Q,QR,R
     1,YAW,R0,X0,/NESS/,NCHAR)
      REAL Q(30,60),QR(NR,NNZ),R(30,60)
      COMMON/AREA1/DIST,PITCH,SIZE,MODE,NGN,NIR,NIQ,NIL
      CALL FILT3D(Q,NS,NZ)
      CALL RORG(Q,QR,R,NS,NZ,NR,NNZ,NBOUND)
      K=3
      SC=1.0
      CALL CPLT3D(QR,NR,NNZ,K,DIST,YAW,PITCH,SIZE,MODE,NGN,SC
     1,NIR,NIQ,NIL,R0,X0,NESS,NCHAR)
      CALL ORG(SIZE,SIZE)
      RETURN
      END
      SUBROUTINE LASPLT(P,NZ,N0,T0,R0,L0,NBOUND)
C**********************************************************************
C     *               PLOT LASER POWER VERSUS Z.  - IN GW.           *
C     *                      MAY,23,1977.                            *
C     *                    RICHARD D. MILROY                         *
C**********************************************************************
      REAL P(120),Z(120),N0,L0
      N=NZ+1
      IF(NBOUND.EQ.2) N=2*NZ-2
      NP1=N+1
      NP2=N+2
      XLEN=6.0
      YLEN=4.0
      DO 2 I=1,N
    2 P(I)=4.92E-22*N0*T0**1.5*R0*R0*P(I)
      AXDELT=NBOUND/XLEN*L0
      DTIC=XLEN/8.0
      CALL AXIS2(0.0,0.0,'Z',-1,XLEN,0.0,0.0,AXDELT,DTIC)
      CALL SCALE(P,YLEN,N,1)
      YMIN = P(NP1)
      YDELT = P(NP2)
      CALL AXIS2(0.,0.,'LASER POWER (GW)',16,YLEN,90.,YMIN,YDELT,-1.)
      DZ=1.0/(NZ-NBOUND)
      DO 1 I=1,N
    1 Z(I)=(I-1.5)*DZ
      Z(NP1)=0.0
      Z(NP2)=NBOUND/XLEN
      CALL LINE(Z,P,N,1,0,1)
      CALL ORG(XLEN,YLEN)
      RETURN
      END
      SUBROUTINE RBND(SIZE,RX1,RX,Z,NS,NZ,NZ2,L0,R0)
      REAL RX1(30,60),RX(NZ2),L0,Z(NZ2)
C-------- PLOT R(S,Z) : SHELL CENTERS.
      XLEN = SIZE
      YLEN = SIZE/2.0
      XDELT=1.0/XLEN*L0
      XDTIC=XLEN/4.0
      YDTIC=-YLEN/4.0
      CALL AXIS2(0.0,0.0,'Z',-1,XLEN,0.0,0.0,XDELT,XDTIC)
      CALL AXIS2(0.0,YLEN,' ',1,XLEN,0.0,0.0,XDELT,XDTIC)
      YDELT=1.0/YLEN*R0
      CALL AXIS2(0.0,0.0,' ',1,YLEN,90.0,0.0,YDELT,YDTIC)
```

```
      HT=0.15
      VX=-0.8
      VY=(YLEN/2.0)-0.5*HT
      CALL SYMBOL(VX,VY,HT,'R',0.0,1)
      CALL AXIS2(XLEN,0.0,' ',-1,YLEN,90.0,0.0,YDELT,YMTIC)
      RX(NX+1)=0.0
      RX(NX+2)=1.0/YLEN
C-------- DEFINE X - VECTOR.
      XS=1.0/NX
      X(1)=0.5*XS
      DO 6 I=2,NX
    6 X(I)=X(I-1)+XS
      X(NX+1)=0.0
      X(NX+2)=1.0/XLEN
      DO 7 I=1,NS
      DO 8 J=1,NX
    8 RX(J)=RX1(I,J)
    7 CALL LINE(X,RX,NX,1,0,1)
      NCHAR=16
      VY=-0.8
      VX=(XLEN/2.0)-0.5*NCHAR*HT
      CALL SYMBOL(VX,VY,HT,'SHELL BOUNDARIES',0.0,NCHAR)
      CALL ORG(SIZE,SIZE)
      RETURN
      END
C     SUBROUTINE CPLT3D(A,N,M,K,DIST  ,YAW,PITCH,SIZE,KODE,NCN,SCAL)
C  DRAW A PERSPECTIVE VIEW OF A CONTOURED SURFACE.
C
C  THIS SET OF SUBROUTINES  (6 OF THEM), HAVING THE NAMES:
C  CPLT3D, AUX060, AUX061, AUX062, AUX063, AND AUX064, WERE GIVEN
C  TO STANFORD BY HOWARD JESPERSON OF IOWA STATE UNIVERSITY.
C
C  THEY HAVE BEEN EXTENSIVELY MODIFIED BECAUSE:
C       THE NAMES OF OUR PLOTTING SUBROUTINES ARE DIFFERENT.
C       THE METHOD OF SCALING HAS BEEN CHANGED.
C       SOME NEW FEATURES HAVE BEEN ADDED.
C       SOME OLD FEATURES HAVE BEEN MADE OPTIONAL.
C
C  IN SPITE OF THE AMOUNT OF MODIFICATION, THE ALGORITHM USED TO
C  DO THE PROJECTION, AND TO DETERMINE THE VISIBILITY OF THE POINTS
C  REFLECTS THE WORK DONE AT IOWA.
C
C  MODIFIED FOR USE AT STANFORD BY: ROBERT J. BEEBE, CAMPUS FACILITY,
C  STANFORD UNIVERSITY.
C
C  DATE OF LAST REVISION: MAY 1, 1969
C
C  A IS THE 2-DIMENSIONED ARRAY CONTAINING THE
C     FUNCTION VALUES.
C
C  WARNING... THE CONTENTS OF THIS ARRAY ARE TRANSFORMED INSIDE THESE
C             ROUTINES (SOME PEOPLE WOULD USE THE TERM "DESTROYED").
C
C  N IS THE NUMBER OF ROWS IN THE ARRAY A.
C
C  M IS THE NUMBER OF COLUMNS IN THE ARRAY A.
C
C  K IS A CODE THAT TELLS WHETHER TO DRAW THE GRID LINES:
C      K=1: ALONG THE N-DIMENSION ONLY.
C      K=2: ALONG THE M-DIMENSION ONLY.
```

```
C       N=3: ALONG BOTH DIMENSIONS.
C
C  SDISTS TELLS HOW FAR AWAY THE SURFACE IS FROM YOUR EYE.
C      THE UNIT OF MEASUREMENT FOR THIS DIMENSION IS THE
C      DIAGONAL OF A CUBE THAT ENCLOSES THE SURFACE.
C      THIS DOES NOT CHANGE THE SIZE OF THE PICTURE, BUT ONLY
C      THE DISTORTION.
C        SDISTS >6 USUALLY WON'T SHOW ANY DISTORTION DUE TO
C                PARALAX.
C        SDISTS <1 MAY CAUSE UNPREDICTABLE SCALING ERRORS.
C
C  YAW (IN DEGREES) SHOWS HOW THE OBJECT IS TURNED AWAY FROM
C  THE VIEWER.
C      POSITIVE YAW WILL TEND TO BRING THE RIGHT EDGE INTO VIEW.
C      ZERO YAW SHOWS THE SURFACE WITH A(1,1) AT THE LEFT FRONT CORNER,
C      A(1,M) AT THE LEFT REAR CORNER, AND A(N,1) AT THE RIGHT FRONT.
C
C  PITCH (IN DEGREES) SHOWS HOW THE SURFACE IS LOWERED OR RAISED AT THE
C  FRONT EDGE.
C      POSITIVE PITCH TENDS TO EXPOSE THE UPPER SURFACE.
C      IF THE MAGNITUDE OF PITCH EXCEEDS 90 DEGREES, FUNNY THINGS
C      HAPPEN TO THE ORIENTATION OF THE PICTURE.
C
C  SIZE (IN INCHES) TELLS HOW LARGE TO MAKE THE PROJECTION OF THE "CUBE"
C  THAT ENCLOSES THE SURFACE.
C     THIS "CUBE" THAT WE HAVE BEEN TALKING ABOUT HAS THE FOLLOWING
C     DIMENSIONS: N,M,MAX(N,M). THE FUNCTION VALUES ARE SCALD TO FIT
C     INSIDE THIS CUBE BEFORE ANY ROTATION IS DONE.
C     AFTER ROTATION, THE CUBE IS SCALD TO FIT INSIDE
C     THE SQUARE PROJECTION PLANE (SIZE BY SIZE). BOTH
C     HORIZONTAL, AND VERTICAL SCAL FACTORS ARE COMPUTED,
C     AND BOTH ARE SET EQUAL TO THE SMALLER OF THE TWO.
C     THIS MEANS THAT, ALTHOUGH THE PROJECTION MAY NOT ALWAYS
C     APPEAR TO BE THE SAME SIZE, IT WILL ALWAYS HAVE THE
C       SAME SHAPE.
C
C  KODE TELLS WHETHER TO DRAW THE "HIDDEN" LINES:
C     KODE=0: ASSUME THE SURFACE IS OPAQUE, SO DON'T DRAW THE
C             "HIDDEN" LINES.
C     KODE=1: ASSUME THE SURFACE IS TRANSPARENT, SO ALL THE LINES
C             ARE PLOTTED.
C
C  MGN TELLS WHETHER TO DRAW THE OUTLINE OF THE CUBE TO HELP
C     ORIENT THE VIEWER.
C    MGN=0: DO NOT DRAW THE OUTLINE OF THE CUBE AT ALL.
C    MGN=1: DRAW THE OUTLINE OF THE CUBE, BUT PUT IT IN ITS OWN
C           TWO-INCH FRAME JUST TO THE LEFT OF THE SURFACE PLOT.
C           DO NOT DRAW THIS CUBE FULL SIZE, BUT MAKE IT ABOUT
C           TWO INCHES ACROSS.
C    MGN=2: DRAW THE OUTLINE OF THE CUBE SUPERIMPOSED ON THE
C           SURFACE PLOT. CPLT3D WILL NOT HIDE ANY OF THE EDGES
C           OF THE CUBE, REGARDLESS OF WHETHER OR NOT ANY OF THE
C           LINES IN THE SURFACE PLOT ARE HIDDEN.
C    MGN=3: DRAW ONLY THE THREE EDGES OF THE CUBE THAT MEET AT THE
C           ORIGIN, SUPERIMPOSED ON THE SURFACE PLOT.
C
C
C  SCAL TELLS THE ROUTINES HOW TALL THEY SHOULD MAKE THE SURFACE,
C  RELATIVE TO THE HEIGHT OF THE CUBE.
C
```

```
C     SCAL=0.: DO NOT SCAL THE DATA AT ALL, BUT TRUST THE USER THAT
C              THE DATA IS NOT SO HIGH THAT IT RUNS OFF THE PAPER.
C
C     SCAL=1.: SCAL THE DATA SO THE TOP OF THE DATA JUST TOUCHES
C              THE TOP OF THE CUBE.
C
C     SCAL=.3: SCAL THE DATA SO THE TOP OF THE SURFACE IS
C              THREE TENTHS AS HIGH AS THE CUBE.
C
C     SCAL=4.: SCAL THE DATA SO THE TOP OF THE SURFACE IS
C              FOUR TIMES AS HIGH AS THE CUBE. (DANGEROUS)
C
C   WARNING... IT IS VERY EXPENSIVE TO DRAW OPAQUE SURFACES, BECAUSE
C              THE PROGRAM HAS TO DETERMINE THE VISIBILITY OF EVERY
C              POINT. THE COMPUTER TIME DOUBLES OR TRIPLES, DEPENDING
C              ON HOW MANY LINE SEGMENTS ARE PARTIALLY VISIBLE.
C
C
C          ... THIS IS NOT A STAND-ALONE PACKAGE, BUT IT IS INTENDED
C              THAT THIS SUBROUTINE(S) WILL BE USED ALONG WITH OTHER
C              SUBROUTINES FROM THE CALCOMP PLOTTING PACKAGE.
C              THE OUTPUT ROUTINES MUST BE INITIALIZED BEFORE USING
C              THIS PACKAGE. SEE THE WRITEUP FOR PROGRAM NUMBER C023.
C
C          ... DO NOT USE SUBROUTINES, OR NAMED COMMON, WITH ANY
C              OF THE FOLLOWING NAMES:
C                CPLT3D, AUX060, AUX061, AUX062, AUX064
C                (THESE ARE SUBROUTINES IN THE CPLT3D PACKAGE)
C                    --OR--
C                COM024, COM025
C                (THESE ARE FOR NAMED COMMON USED BY THE CPLT3D
C                PACKAGE).
C
C          ... DO NOT FORGET, THE CONTENTS OF THE ARRAY A GET CLOBBERED.
      SUBROUTINE CPLT3D(A,N,M,K,DIST,YAW,PITCH,SIZE,KODE,NGN,SCAL
     1,NIR,NIQ,NIL,RO,XO,/MESS/,NCHAR)
       COMMON /COM024/ ANGA , ANGB , HV , D, SH,SV
       COMMON /COM025/SL , SM , SN , CX , CY , CZ , QX , QY , QZ , SD
      DIMENSION H( 10 ) , V( 10 ) , X(2) , Y(2) , Z(2) , XP( 8 ),
     :     A (N,M),DZ(4)
C***** ********
      SDISTS=DIST
      ANGA = (YAW+270.) * .0174532
      ANGB = PITCH * .0174532
      HV = SIZE
C  DIRECTION COMPONENTS TO THE EYE.
      SL = -COS( ANGA ) * COS( ANGB )
      SM = -SIN( ANGA ) * COS( ANGB )
      SN = -SIN ( ANGB )
           IF ( ABS( SN )  .NE.  1.0  )    GO TO 10
           WRITE(  6      ,      20  )
   20  FORMAT(  '1' , 20X ,  20('*') , / '0', 'YOU ARE ATTEMPTING TO LOO
     :K  STRAIGHT DOWN ( OR UP ) AT THE SURFACE ' )
            GO TO 2150
   10  CONTINUE
      SD = 1.0 / SQRT( 1.0 - SN ** 2  )
      X(1) = 1
      X(2) = N
      Y(1) = 1
      Y(2) = M
```

```
      T=MIN0(M,N)
C  FIND THE DIAGONAL OF THE "CUBE".
       D = M ** 2  +  N ** 2  +  T ** 2
       D  =  SQRT ( D )
       SCL = SDISTS * D
C  COORDINATES OF YOUR EYE.
       CX = -SL * SCL
       CY = -SM * SCL
       CZ = -SN * SCL
C  COORDINATES OF THE PROJECTION PLANE.
       QX = CX + D * SL
       QY = CY + D * SM
       QZ = CZ + D * SN
 2060 CALL AUX060 ( A, N, M, Z ,KODE)
C
      DZ(1) = Z(1)
      DZ(2) = Z(2)
      VUQ = NIQ
      CALL SCALE(DZ,VUQ,2,1)
      Z(1) = DZ(3)
      Z(2) = VUQ*DZ(4) + Z(1)
C
      RANGE= (Z(2)-Z(1))
      S=1.
      IF( SCAL .NE. 0 ) S=T/RANGE*SCAL
C  SCAL THE SURFACE TO MAKE A "CUBE".
       DO 30 I = 1 , N
       DO 30 J = 1 , M
       A ( I , J ) = ( A ( I , J ) - Z ( 1 )  ) * S
   30 CONTINUE
      QMIN=Z(1)
      QMAX=Z(2)
       Z(1) = 0.0
       Z(2) = T
 2080 CALL AUX061   ( X, Y, Z, XP , N , V ,KODE)
       DO  2130 I = 1 , 8
       H( I ) = ( (XP(I) - QX ) * SM - (  H(I) - QY ) * SL ) * SD
       V ( I ) = (  V( I ) - QZ ) * SD
 2130 CONTINUE
 2100 CALL AUX060 ( H, 8, 1, H(9) ,KODE)
 2120 CALL AUX060 ( V, 8, 1, V(9) ,KODE)
      IF( MGN .EQ. 0) GO TO 2140
      S=HV
      IF(MGN .EQ. 1) S=1.5
      SH = S/ (H(10)-H(9) )
      SV = S/ (V(10)-V(9) )
      SH = SIGN( AMIN1(SH,SV),SH )
      SV = SIGN(SH,SV)
      IF(MGN.EQ.1) CALL PLOT(0.0,2.0,-3)
      CALL AXDR(H,V,SH,SV,MESS,NCHAR,TAN,NIR,NIQ,NIL,RO,XO,QMIN,QMAX)
 2139  IF(MGN .NE. 1) GO TO 2140
      CALL PLOT (AINT((H(10)-H(9))*SH+2.),-2.05,-3)
 2140 CALL AUX063  ( X, Y, A, N , M , H , V , K ,KODE)
 2150 CONTINUE
         RETURN
      END
C
      SUBROUTINE AUX060 ( A, N, M, Z ,KODE)
      DIMENSION Z(1), A(N, ...
C  FIND THE MAX, AND MIN O...
```

```
C
 1050 Z(1) = A( 1, 1 )
 1060 Z(2) = Z(1)
C
 1080  DO 1190 J = 1 , N
C
 1100   DO 1180 I = 1, N
C
      Z(1) =  AMIN1( Z(1), A(I,J))
      Z(2) = AMAX1( Z(2), A(I,J))
C
 1180   CONTINUE
 1190  CONTINUE
 1230        RETURN
                END
      SUBROUTINE AUX061 ( X, Y, Z, XP , H , V ,KODE)
C  FIND THE CORNERS OF THE ROTATED CUBE.
C
      DIMENSION X(1),Y(1),Z(1),H(1),V(1),XP(1)
C
  050  L = 0
  070  DO 180 I = 1, 2
C
  090   DO 170 J = 1, 2
C
  110    DO 160 K = 1, 2
C
  130     L = L + 1
  140     CALL AUX062 ( X(I), Y(J), Z(K), XP( L ),
     1          H(L) , V( L ),KODE )
  160    CONTINUE
  170   CONTINUE
  180  CONTINUE
  190 RETURN
      END
      SUBROUTINE AUX062 ( X, Y, Z, XP , YP , ZP ,KODE)
C  FIND THE LOCATION OF A POINT IN THE ROTATED CUBE.
       COMMON /COM024/ ANGA , ANGB , HV , D, SH,SV
       COMMON /COM025/SL , SM , SN , CX , CY , CZ , QX , QY , QZ , SD
       SK = D / (  (X - CX ) * SL + ( Y - CY ) * SM + ( Z - CZ ) *SN)
       XP = CX + SK * ( X - CX )
       YP = CY + SK * ( Y - CY )
       ZP = CZ + SK * ( Z - CZ )
             RETURN
                          END
      SUBROUTINE AUX063 (X,Y,A,N,M,H,V,K,KODE)
C  DRAW THE FIGURE.
       COMMON /COM024/ ANGA , ANGB , HV , D, SH,SV
       COMMON /COM025/SL , SM , SN , CX , CY , CZ , QX , QY , QZ , SD
C
      DIMENSION X(1),Y(1), H(1),V(1),A(N,M)
      INTEGER UP , DOWN , PEN , P , Q
        INTEGER P1 , PO
C
C
C
          END = 1.0 / 16.0
C   CAN USE 1 / 32 OR  1 / 64  FOR FINER INTERPOLATION
C
C
```

```
      UP    = 3
      DOWN  = 2
      SH =  HV  / ( H ( 10 ) - H ( 9 ) )
      SV =  HV  / ( V ( 10 ) - V ( 9 ) )
      SH = SIGN(AMIN1(SH,SV),SH)
      SV = SIGN(SH,SV)
      MH = H
      MN = N
  080 IF ( K - 1 ) 100 , 120, 100
C
  100 IF ( K - 3 ) 1110, 120, 1110
C
C  DRAW LINES ALONG THE Y-AXIS
  120 CONTINUE
      L = 0
      LD = 1
        DD = 0.5 *  LD
C
  140  DO 1060 J = 1, N
      Q     = 0
      YJ = J
  160   DO 1030 I = 1, MN
C
           L = L + LD
       XI = L
      CALL AUX064 ( A ,XI ,YJ , M , N , P ,KODE)
       PEN = UP
           IF ( P ) 510 , 520 , 530
  510 CONTINUE
         IF ( Q ) 540 , 550 , 540
  520 CONTINUE
         IF ( Q ) 610 , 1020 , 610
  530  CONTINUE
         IF ( Q ) 540 , 550 , 540
  540  CONTINUE
        PEN = DOWN
       GO TO 170
  550  CONTINUE
          IF ( I   .EQ.   1  )   GO TO 170
          DI = DD
      TO = L - LD
        T = TO + DI
        P1 = Q
  560  IF (  ABS( DI )   .LT.   END  )   GO TO  570
       CALL AUX064 ( A , T , YJ , M , N , PO ,KODE)
        DI = DI * 0.5
       IF ( PO  .EQ.  0  )   GO TO 565
         TO = T
        P1 = PO
        T =  T - DI
        GO TO 560
  565  T = T + DI
        GO TO 560
  570  CONTINUE
       T = TO
      IF ( P1 * P  )    170 , 170 , 580
  580 CONTINUE
  590 CONTINUE
      ZP = A(L-LD,J)+(T-L+LD)*(A(L,J)-A(L-LD,J))/LD
      CALL AUX062(T,YJ,ZP,XP,MH,YV,KODE)
```

```
      HH = ( ( XP-QX)*SH- (HH - QY )*SL ) * SD
      VV =  ( VV - QZ ) * SD
        HH = ( HH - H(9) ) * SH
        VV = ( VV - V(9) ) * SV
       CALL PLOT ( HH , VV , PEN )
 600  PEN = 5 - PEN
      GO TO 170
 610 CONTINUE
        PEN = DOWN
         DI = DD
       TO = L - LD
      T = TO + DI
          P1 = Q
 620  IF ( ABS( DI )  .LT.  END  )  GO TO  630
       CALL AUX064 ( A , T , YJ , N , M , PO ,KODE)
       DI = DI * 0.5
       IF ( PO  .EQ.  0  )   GO TO  625
        TO = T
      P1 = PO
      T = T + DI
       GO TO 620
 625   T = T - DI
         GO TO  620
 630   CONTINUE
       T = TO
         IF (  P1 * Q )   600 , 600 , 590
 170     CALL AUX062 ( XI, YJ, A( L, J ), XP , HH ,VV ,KODE)
      VV =  ( VV - QZ ) * SD
      HH = ( ( XP-QX)*SH- (HH - QY )*SL ) * SD
 190    HH = ( HH - H(9) ) * SH
 200    VV = ( VV - V(9) ) * SV
      CALL PLOT (  HH , VV , PEN )
1020  Q = P
1030  CONTINUE
C
C
      L = L + LD
      LD = -LD
         DD =-DD
C
 1060 CONTINUE
C
C
 1090  IF ( K - 3 ) 2060, 1110 , 2060
C
C  DRAW LINES ALONG THE X-AXIS.
 1110 CONTINUE
C
      L = 0
      LD = 1
        DD =  0.5 * LD
 1140  DO 2040 I = 1 , N
         XI = I
       Q = 0
 1160    DO 2020 J = 1 , MM
             L = L + LD
           YJ = L
       CALL AUX064 ( A ,XI ,YJ , M , N , P ,KODE)
       PEN = UP
             IF ( P ) 1510 , 1520 , 1530
```

```
1510 CONTINUE
       IF ( Q ) 1540 , 1550 , 1540
1520  CONTINUE
      IF ( Q )  1610 , 2010 , 1610
1530  CONTINUE
       IF ( Q ) 1540 , 1550 , 1540
1540   CONTINUE
        PEN = DOWN
       GO TO 1170
1550        CONTINUE
        IF ( J  .EQ.  1  )   GO TO 1170
         DI = DD
     TO = L - LD
       T = TO + DI
       P1 = Q
1560  IF (  ABS( DI )   .LT.   END  )  GO TO 1570
      CALL AUX064 ( A , XI , T , N , M , PO ,KODE)
       DI = DI * 0.5
      IF ( PO  .EQ.  0  )  GO TO 1565
        TO = T
       P1 = PO
       T =  T - DI
         GO TO 1560
1565  T = T + DI
          GO TO 1560
1570    CONTINUE
       T = TO
      IF ( P1 * P  )  1170 , 1170 , 1580
1580  CONTINUE
1590  CONTINUE
    ZP=A(I,L-LD) + (T-L+LD) * (A(I,L) - A(I,L-LD))/LD
         CALL AUX062 ( XI , T , ZP , XP,HH,VV ,KODE)
      HH = ( ( XP-QX)*SH-  (HH - QY )*SL ) * SD
      VV =   ( VV - QZ ) * SD
        HH = ( HH - H(9) ) * SH
        VV = ( VV - V(9) ) * SV
       CALL PLOT ( HH , VV , PEN )
1600 PEN = 5 - PEN
       GO TO 1170
1610  CONTINUE
        PEN = DOWN
         DI = DD
       TO = L - LD
      T = TO + DI
          P1 = Q
1620  IF ( ABS( DI )   .LT.   END  )  GO TO 1630
       CALL AUX064 ( A , XI , T , N , M , PO ,KODE)
       DI = DI * 0.5
       IF ( PO  .EQ.  0  )   GO TO 1625
        TO = T
      P1 = PO
      T = T + DI
       GO TO 1620
1625   T = T - DI
         GO TO 1620
1630    CONTINUE
       T = TO
       IF ( P1 * Q ) 1600 , 1600 , 1590
1170    CALL AUX062 ( XI, TJ, A( I, L ), XP , HH ,VV ,KODE)
      HH = ( ( XP-QX)*SH-  (HH - QY )*SL ) * SD
```

```
      VV =  ( VV - QZ ) * SD
 1180    HH = ( HH - H(9) ) * SH
 1190    VV = ( VV - V(9) ) * SV
        CALL PLOT (  HH , VV ,  PEN  )
 2010  Q = P
 2020   CONTINUE
C
       L = L + LD
       LD = - LD
          DD =-DD
 2040  CONTINUE
C
 2060 CONTINUE
C
 2130  RETURN
        END
      SUBROUTINE AUX064 (Z, XI, YJ, M, N, P,KODE)
C  SEE IF A POINT IS VISIBLE.
       COMMON /COM024/ ANGA , ANGB , HV , D, SH,SV
       COMMON /COM025/SL , SM , SN , CX , CY , CZ , QX , QY , QZ , SD
       INTEGER  CUM , CNT , P
         REAL I , J , II , JJ
       DIMENSION   Z(M,N)
      IF( KODE .EQ. 1) GO TO 78
      IR = XI
      JC = YJ
        ZB = Z ( IR , JC )
      IF (  XI   .EQ.  IR  )  GO TO 2
       ZB = Z(IR , JC ) + ( XI - IR ) * ( Z(IR + 1 , JC ) - Z( IR ,JC))
           GO TO 4
    2  IF ( YJ  .EQ.  JC )       GO TO 4
      ZB = Z(IR , JC) + (YJ-JC)*(Z(IR,JC+1) - Z(IR,JC ))
    4  CONTINUE
       XEND = 0.0
       DX = 0.0
       YMULT = 0.0
       ZMULT = 0.0
         IF (XI    .EQ.   CX  )  GO TO 10
      YMULT = (YJ - CY )  /  (XI - CX )
         ZMULT = ( ZB - CZ ) / ( XI - CX )
      DX = 1.0
        XEND = M + 1
        IF ( XI   .LT.  CX )  GO TO 10
        DX = -1.0
       XEND = 0.0
   10  CONTINUE
      YEND = 0.0
       DY = 0.0
        XMULT = 0.0
        IF ( YJ  .EQ.  CY  )  GO TO 20
       XMULT = ( XI - CX ) / (YJ - CY )
        IF (  ZMULT  .EQ.  0.0  )  ZMULT=(ZB - CZ ) / ( YJ - CY )
       DY = 1.0
       YEND = N + 1
       IF ( YJ  .LT.  CY )  GO TO 20
       DY = -1.0
       YEND = 0.0
   20 CONTINUE
      CUM = 0
      CNT = 0
```

```
      P = 0
       XB = XI
       YB = YJ
   30 CONTINUE
      II = AINT( XB )
      JJ = AINT( YB )
      XSTEP = DX
      YSTEP = DY
        IF ( XB  .EQ.  II )  GO TO 40
        IF ( DX  .LT.  0.0 )  XSTEP = 0.0
      GO TO 45
   40   IF ( YB  .EQ.  JJ )  GO TO 45
        IF ( DY  .LT.  0.0 )  YSTEP = 0.0
   45 CONTINUE
        I = II + XSTEP
        J = JJ + YSTEP
        IF ( I  .EQ.  XEND )  GO TO 80
        IF ( J  .EQ.  YEND )  GO TO 80
      XB = CX + XMULT * ( J - CY )
      YB = CY + YMULT * ( I - CX )
        IF ( DX  .LT.  0.0 )  GO TO 55
        IF ( XB  .LT.  I  )  GO TO 60
   50 XB = I
       GO TO 65
   55  IF ( XB  .LT.  I )  GO TO 50
   60 YB = J
   65  CONTINUE
      ZB = CZ + ZMULT * ( XB - CX )
      IR = I
      JC = J
        IF ( YB  .NE.  J )  GO TO 70
       IDX = I - DX
      ZS = Z( IR, JC ) - DX * ( XB - I ) * (Z(IDX,JC) - Z(IR,JC))
      GO TO 75
   70 JDY = J - DY
      ZS = Z( IR,JC ) - DY * ( YB-J ) * (Z( IR,JDY ) - Z( IR,JC ) )
   75 CONTINUE
      SGN = 1
      IF (  ZB  .LT.  ZS  )  SGN = -1
      CUM = CUM + SGN
      CNT = CNT + 1
      IF (  IABS ( CUM )   .EQ.   CNT  )  GO TO 30
       GO TO 90
78    P=1
      GO TO 95
   80  CONTINUE
      P = 1
          IF ( CUM ) 84 , 86 , 90
   84 P = -1
              GO TO 90
   86  CONTINUE
               IF ( ZB  .LE.  CZ  )  GO TO 90
       P = -1
   90  CONTINUE
95      RETURN
      END
      SUBROUTINE AXDR(H,V,SH,SV,/NESS/,NCHAR,YAN,NIR,NIQ,NIL,X0,Y0
     1,QMIN,QMAX)
      DIMENSION H(10),V(10),X(8),Y(8)
C------- PUT THE COORDINATES OF EACH OF THE 8 CUBE CORNERS IN X&Y.
```

```
      DO 1 I=1,8
      X(I) = (X(I)-X(9))*SX
      Y(I) = (Y(I)-Y(9))*SY
    1 CONTINUE
      CALL PLOT(X(2),Y(2),3)
      CALL PLOT(X(4),Y(4),2)
      CALL PLOT(X(3),Y(3),2)
      CALL PLOT(X(7),Y(7),2)
      CALL PLOT(X(8),Y(8),2)
      CALL PLOT(X(4),Y(4),2)
      CALL PLOT(X(8),Y(8),3)
      CALL PLOT(X(6),Y(6),2)
      CALL PLOT(X(2),Y(2),2)
C------- DRAW THE RADIUS AXIS.
      CALL AXPL(1,5,X,Y,0.0,RO,'RADIUS',-6,NIR,1.0)
      IF(YAW.GT.180.0) GO TO 2
C------- DRAW THE APPROPRIATE AXIS FOR 0.LT.YAW.LE.45.
      CALL AXPL(5,7,X,Y,0.0,ZO,'Z',-1,NIL,1.0)
      CALL AXPL(1,2,X,Y,QMIN,QMAX,MESS,NCHAR,NIQ,-1.0)
C------- DRAW LINES TO FINISH BOX.
      CALL PLOT(X(5),Y(5),3)
      CALL PLOT(X(6),Y(6),2)
      CALL PLOT(X(1),Y(1),3)
      CALL PLOT(X(3),Y(3),2)
      RETURN
    2 CONTINUE
C------- DRAW THE APPROPRIATE AXIS FOR 325.LT.YAW.LT.360.
      CALL AXPL(3,1,X,Y,ZO,0.0,'Z',-1,NIL,1.0)
      NCH = -NCHAR
      CALL AXPL(5,6,X,Y,QMIN,QMAX,MESS,NCH,NIQ,-1.0)
C------- DRAW LINES TO FINISH BOX.
      CALL PLOT(X(5),Y(5),3)
      CALL PLOT(X(7),Y(7),2)
      CALL PLOT(X(1),Y(1),3)
      CALL PLOT(X(2),Y(2),2)
      RETURN
      END
      SUBROUTINE AXPL(I,J,X,Y,QMIN,QMAX,/MESS/,NCHAR,NIQ,DTS)
      DIMENSION X(8),Y(8)
      DX = X(J)-X(I)
      DY = Y(J)-Y(I)
      AXLEN = SQRT(DX**2 +DY**2)
      ANGLE = 57.296*ATAN(DY/DX)
      AXMIN = QMIN
      AXDELT = (QMAX-QMIN)/AXLEN
      DTIC = AXLEN/NIQ*DTS
C     CALL AX2EP(DTIC,2,1,0,D,D,D,D,D,D,D,D)
      CALL AXIS2(X(I),Y(I),MESS,NCHAR,AXLEN,ANGLE,AXMIN,AXDELT,DTIC)
      RETURN
      END
      SUBROUTINE FILT3D(Q,NS,NX)
C*****************************************************************
C  *      FILTER ANY JITTER FROM OUTPUT OF 2-D AND ROUTINE.       *
C  *               RICHARD D. KILROY        78-02-16              *
C*****************************************************************
      REAL Q(30,60)
      INTEGER S,X
C------- FILTER IN X-DIRECTION
      NXM1 = NX - 1
      DO 1 S=1,NS
```

```
      Q2 = Q([illegible],1)
      DO 1 J=2,[illegible]
      Q1 = Q([illegible])
      Q([illegible]) = [illegible]*Q([illegible]) + 0.25*(Q2+Q([illegible],J+1))
      Q2 = Q1
    1 CONTINUE
C------- FILTER IN I-DIRECTION.
      NIM1 = NI - 1
      DO 2 J=1,NJ
      Q2 = Q(1,J)
      DO 2 I=2,NIM1
      Q1 = Q(I,J)
      Q(I,J) = 0.5*Q(I,J) + 0.25*(Q2+Q(I+1,J))
      Q2 = Q1
    2 CONTINUE
      RETURN
      END
```

```
C**************************************************************************
C    *      DRAW SOME GRAPHS FOR 2-D SIMULATION.                          *
C    *      RICHARD D. [illegible]                   78-02-09             *
C**************************************************************************
      REAL A(30,60),TE(30,60),TI(30,60),N(30,60),PHI(31),V(30,60)
     1     ,VPS(30,60),P(120),E(30,60)
     2     ,NT(500),TET(500),TIT(500),TT(500)
     3     ,N1(500),TE1(500),TI1(500),X(500)
     4     ,NMAX,L0,N0,VPLT(100,10)
      INTEGER XINT,RINT,TINT
      LOGICAL RDN
      XLEN = 4.0
      YLEN = 5.0
      TIN = 5.E-7
      TINC = 5.E-7
      TFIN = 30.E-6
      TIN0 = 1.5E-6
      NMAX = 5.E+17
      NINT = 5
      XINT = 5
      RINT = 3
      TINT = 5
      X0 = 50.0
      READ(3) NTAPE,NTIME,T,NS,NX,L0,R0,N0,T0,DT,NBOUND,XLAS
     1,XMIN,RDN,RM
      BACKSPACE 3
      V0 = SQRT(9.58E+11*T0)/SQRT(RM)
      TN0 = L0/SQRT(9.58E+11*T0)*SQRT(RM)
      DX = L0/(NX-NBOUND)
      KX = X0/DX + 0.5
      KX = NX+1-KX
      IT = 0
      KK = 0
      T = -1.0E+50
      CALL PLOTS
      CALL ORG1(XLEN,YLEN)
    2 KK =KK+1
      IF(T.GT.TFIN) GO TO 98
      TD = TIN + (KK-1)*TINC
    1 TOLD = T
      READ(3,END=99) NTAPE,NTIME,T,NS,NX,L0,R0,N0,T0,DT,NBOUND
      T = T*TN0
      IF(T.LT.1.E-30) T=1.0E-30
      IF(T.GT.TD) GO TO 3
      READ(3) A
      READ(3) TE
      READ(3) TI
      READ(3) N
      READ(3) PHI
      READ(3) V
      READ(3) P
      READ(3) VPS
      READ(3) E
      IT = IT+1
      NT(IT) = N(1,KX)*N0
      TET(IT) = TE(1,KX)*T0
      TIT(IT) = TI(1,KX)*T0
      TT(IT) = T
      GO TO 1
```

```
    3 CONTINUE
      IF (ABS(TD-T) .GT. ABS(TD-TOLD)) GO TO 4
      READ(3) A
      READ(3) TE
      READ(3) TI
      READ(3) N
      READ(3) PHI
      READ(3) V
      READ(3) P
      READ(3) VPS
      READ(3) U
      IT = IT+1
      NT(IT) = N(1,KX)*N0
      TET(IT) = TE(1,KX)*T0
      TIT(IT) = TI(1,KX)*T0
      TT(IT) = T
      GO TO 5
    4 BACKSPACE 3
      T = TOLD
      GO TO 5
   99 TFIN = 0.0
    5 CONTINUE
C------- PLOT CURRENT TIME IN SECONDS.
      TIME=T
      NEXP=ALOG10(TIME)
      NEXP=NEXP-1
      EXP=NEXP
      BASE=TIME/(10.0**NEXP)
      HT=(YLEN-2.0)/15.0
      CALL SYMBOL(1.0,1.0,HT,'TIME=      X 10',0.0,14)
      VX=1.0+5.0*HT
      CALL NUMBER(VX,1.0,HT,BASE,0.0,2)
      VX=1.0+10.0*HT
      VY=1.0+0.5*HT
      HT=0.5*HT
      CALL NUMBER(VX,VY,HT,EXP,0.0,-1)
      CALL ORG(XLEN,[illegible])
C-------  DEFINE VECTORS FOR PLOTTING RADIAL PLOTS.
      AT=0.0
      AT1=0.0
      DO 10 I=1,NS
      AT = AT+A(I,KX)
      R(I) = SQRT(0.5*(AT+AT1))
   10 AT1 = AT
      DO 6 I=1,NS
      X(I) = R(I)*R0
      N1(I) = N(I,KX)*N0
      TE1(I) = TE(I,KX)*T0
    6 TI1(I) = TI(I,KX)*T0
      CALL XPLOT(NS,X,N1,TE1,TI1,XLEN,YLEN,NINT,NMAX,RINT,'R',-1,R0)
C------- DEFINE VECTORS FOR PLOTTING AXIAL PLOTS.
      DO 7 I=NBOUND,NX
      J=NX+1-I
      X(J) = DX*(J-1)
      N1(J) = N(1,I)*N0
      VPLT(J,[illegible]) = V(1,I)*V0
      TE1(J) = TE(1,I)*T0
    7 TI1(J) = TI(1,I)*T0
      CALL XPLOT(NX,X,N1,[illegible],[illegible],XLEN,YLEN,NINT,NMAX,XINT,'Z',-1,L0)
      CALL LABELS(T,NX,N0,T0,[illegible],L0,NBOUND,XLEN,YLEN,XINT,R0)
```

```
      GO TO 2
   98 CONTINUE
C------- MAKE PLOTS AS A FUNCTION OF TIME.
      CALL XPLOT(IT,TT,HT,TET,TIT,XLEN,YLEN,NINT,HMAX,XINT,'T',-1,TZERO)
C------- PLOT AXIAL VELOCITY VS X.
      CALL VPLOT(NZ,X,VPLT,XLEN,YLEN,XINT,KK,L0)
      CALL PLOT(0.0,0.0,999)
      STOP
      END
      SUBROUTINE AXDR(XLEN,YLEN,/XAX/,NTIT,XMIN,XDELT,XDT
     *      ,TMIN,TDELT,TDT,HMIN,HDELT,HDT)
      REAL HMIN,HDELT,HDT
      CALL AXIS2(0.0,0.0,XAX,NTIT,XLEN,0.0,XMIN,XDELT,XDT)
      CALL AXIS2(0.0,0.0,'T',1,YLEN,90.0,TMIN,TDELT,TDT)
      CALL AXIS2(XLEN,0.0,'H',-1,YLEN,90.0,HMIN,HDELT,HDT)
      CALL PLOT(0.0,YLEN,3)
      CALL PLOT(XLEN,YLEN,2)
      RETURN
      END
      SUBROUTINE XPLOT(NP,X,H,TE,TI,XLEN,YLEN,NINT,HMAX,XINT,/XAX/,NTIT
     *      ,X0)
      REAL X(500),H(500),TE(500),TI(500),HMAX,HMIN,HDELT,HDT,SC(4)
      INTEGER XINT
      CALL FILT1D(TE,NP)
      CALL FILT1D(TI,NP)
      CALL FILT1D(H,NP)
      TMIN = 1.E50
      TMAX = -1.E50
      DO 1 I=1,NP
      IF(TE(I).LT.TMIN) TMIN=TE(I)
      IF(TI(I).LT.TMIN) TMIN=TI(I)
      IF(TE(I).GT.TMAX) TMAX=TE(I)
      IF(TI(I).GT.TMAX) TMAX=TI(I)
      IF(H(I).GT.HMAX) H(I) = HMAX
    1 CONTINUE
      SC(1) = TMIN
      SC(2) = TMAX
      CALL SCALE(SC,YLEN,2,1)
      XMIN = 0.0
      XDELT= X0/XLEN
      XDT = XLEN/XINT
      TMIN = SC(3)
      TDELT = SC(4)
      TDT = -1.0
      HMIN = 0.0
      HDELT = HMAX/YLEN
      HDT = -YLEN/NINT
      TE(NP+1) = SC(3)
      TE(NP+2) = SC(4)
      TI(NP+1) = TE(NP+1)
      TI(NP+2) =TE(NP+2)
      H(NP+1) = HMIN
      H(NP+2) = HDELT
      X(NP+1) = XMIN
      X(NP+2) = XDELT
      CALL AXDR(XLEN,YLEN,XAX,NTIT,XMIN,XDELT,XDT
     *      ,TMIN,TDELT,TDT,HMIN,HDELT,HDT)
      CALL LINE(X,TE,NP,1,0,0)
      CALL LINE(X,TI,NP,1,0,0)
      CALL LINE(X,H,NP,1,0,0)
```

```
      [illegible] ORG(XLEN,YLEN)
      [illegible]
      END
      SUBROUTINE ORG1(XLEN,YLEN)
C---------MOVE ORIGIN IN A WAY TO MINIMIZE PLOTTER PAPER WASTE.
C---------FOR TEKTRONIX, RE-SCALE PLOT AND HALT UNTILL RETURN ENTERED.
      CALL [illegible](61)
      CALL [illegible](61)
      F=AMIN1(30.0/XLEN,30.0/YLEN)
      CALL FACTOR(F)
      CALL PLOT(2.0,2.0,-3)
      GO TO 10
    1 CALL PLOT(3.0,3.0,-3)
      X0=0.0
      Y0=0.0
      XL=XLEN
      YL=YLEN
      GO TO 10
      ENTRY ORG(XLEN,YLEN)
C---------ENTER HERE FOR ALL BUT 1'ST CALL TO THIS ROUTINE.
      CALL TCLEAR(62)
      READ(6,601) NOTHG
  601 FORMAT(I5)
      CALL TERASE(62)
      READ(6,601) NOTHG
      GO TO 10
    2 VX=6.0+XL
      VY=6.0+YL
      Y=Y0+VY+YLEN
      YL=YLEN
      IF(Y.GT.33.0) GO TO 3
      CALL PLOT(0.0,VY,-3)
      Y0=Y0+VY
      IF(XLEN.GT.XL) XL=XLEN
      GO TO 10
    3 VY=-Y0
      CALL PLOT(VX,VY,-3)
      XL=XLEN
      X0=X0+VX
      Y0=0.0
   10 RETURN
      END
      SUBROUTINE LASPLT(P,NZ,N0,T0,R0,L0,NBOUND,XLEN,YLEN,XINT,X0)
C*****************************************************************************
C     *                  PLOT LASER POWER VERSUS Z.  - IN GW.                 *
C     *                           MAY,23,1977.                                *
C     *                        RICHARD D. MILROY                              *
C*****************************************************************************
      REAL P(120),Z(120),N0,L0,INTEN(120)
      INTEGER XINT
      NS = 0.25
      RL0 = 0.0035
      N=NZ+1
      IF(NBOUND.EQ.2) N=2*NZ-2
      NP1=N+1
      NP2=N+2
      DO 2 I=1,N
    2 P(I)=4.52E-22*N0*T0**1.5*R0*R0*P(I)
      AKDELF=NBOUND/XLEN*L0
      DTIC=XLEN/XINT
```

```
      CALL AXIS2(0.0,0.0,'X',-[illegible]XLEN,0.0,0.0,AXDELT,DTIC)
      CALL SCALE(P,YLEN,N,1)
      YMIN = P(NP1)
      YDELT = P(NP2)
      CALL AXIS2(0.,0.,'LASER POWER (GW) ',16,YLEN,90.,YMIN,YDELT,-1.)
      DS=1.0/(NS-NBOUND)
      DO 1 I=1,N
    1 X(I)=(I-1.5)*DS
      X(NP1)=0.0
      X(NP2)=NBOUND/XLEN
      CALL LINE(X,P,N,1,0,1)
      CALL PLOT(0.0,YLEN,3)
      CALL PLOT(XLEN,YLEN,2)
C------[illegible]ND PLOT INTENSITY.
      [illegible]
      [illegible] DS*ABS(X0-X(I)*L0)
      [illegible]/(6.283*SIGMA**2)*1.E9
      [illegible]NTEN,[illegible]N,N,1)
      [illegible]NP1)
      [illegible](NP2)
      [illegible]IS2(XLEN,0.0,'INTENSITY',-9,YLEN,90.,YMIN,YDELT,-1.)
      CA[illegible]LINE(X,INTEN,N,1,0,1)
      CALL ORG(XLEN,YLEN)
      RETURN
      END
      SUBROUTINE FILT1D(Q,N)
C**********************************************************************
C     *      FILTER JITTER FROM RESULTS BEFORE PLOTTING.               *
C     *            RICHARD D. MILROY          78-02-14                 *
C**********************************************************************
      REAL Q(1)
      NM1 = N-1
      Q2 = Q(1)
      DO 1 I=2,NM1
      Q1 = Q(I)
      Q(I) = 0.5*Q(I) + 0.25*(Q2+Q(I+1))
      Q2 = Q1
    1 CONTINUE
      RETURN
      END
      SUBROUTINE VPLOT(N,X,VPLT,XLEN,YLEN,XINT,KK,X0)
C**********************************************************************
C     *      PLOT VELOCITY FOR SEVERAL TIMES.                          *
C     *            RICHARD D. MILROY                 78-02-15          *
C**********************************************************************
      REAL VPLT(100,10),V(100),SC(4),X(500)
      INTEGER XINT
      K = KK-1
C------ FIND MAX. AND MIN. OF V AND SCALE.
      VMAX = -1.E50
      VMIN = 1.E50
      DO 1 J=1,K
      DO 1 I=1,N
      IF(VPLT(I,J).LT.VMIN) VMIN = VPLT(I,J)
      IF(VPLT(I,J).GT.VMAX) VMAX = VPLT(I,J)
    1 CONTINUE
      SC(1) = VMIN
      SC(2) = VMAX
      CALL SCALE(SC,YLEN,2,1)
      VMIN = SC(3)
```

```
      VDELT = [illegible](0)
      XMIN = 0.0
      XDELT = X0/XLEN
      XDT = XLEN/XINT
      VDT = -1.0
      CALL AXIS2(0.0,0.0,'X',-1,XLEN,0.0,XMIN,XDELT,XDT)
      CALL AXIS2(0.0,0.0,'V',1,YLEN,90.0,VMIN,VDELT,VDT)
      CALL PLOT(0.0,YLEN,3)
      CALL PLOT(XLEN,YLEN,2)
      CALL PLOT(XLEN,0.0,2)
      DO 2 J=1,K
      DO 3 I=1,N
    3 V(I) = VPLT(I,J)
      CALL FILT1D(V,N)
      V(N+1) = VMIN
      V(N+2) = VDELT
      CALL LINE(X,V,N,1,0,0)
    2 CONTINUE
      RETURN
      END
```

```
C*****************************************************************
C       READ DATA FROM DISK OR TAPE AND PRINT RESULTS.           *
C   *   RICHARD HILBOT  -  JUNE 16, 1977.                        *
C*****************************************************************
      REAL A(30,60),TE(30,60),TI(30,60),N(30,60),PHI(31)
     1,V(30,60),P(120),B(30),R(30),PRESS(30),NO,LO,U(30,60),W(30,60)
      INTEGER X,S,XMIN
      WRITE(6,621)
  621 FORMAT(' INPUT NBCO, NECH, AND SKIP RATE'/'     N     N     N')
      READ(5,501) NBCO,NECH,NSR
  501 FORMAT(3I5)
  101 READ(3,END=102) NTAPE,NTIME,T,M,MX,LO,RO,NO,TO,DT,NR,XLAS,XMIN
      READ(3) A
      READ(3) TE
      READ(3) TI
      READ(3) N
      READ(3) PHI
      READ(3) V
      READ(3) P
      READ(3) W
      READ(3) U
      IF(NTIME.NE.NBCO) GO TO 101
      NBCO=NBCO+NSR
      IF(NTIME.GT.NECH) GO TO 102
      WRITE(7,622) NTIME,T,DT,NR
  622 FORMAT('-NTIME=',I4,'   TIME=',1PE9.2,'   DT=',E9.2,'   NR=',0PI3)
      WRITE(7,623) M,MX,NTAPE,LO,RO,NO,TO
  623 FORMAT(' M=',I3,'  MX=',I3,'  NTAPE,',I3,'  LO=',1PE9.2,'  RO='
     1,E9.2,'  NO=',E9.2,'  TO=',E9.2)
      NNX=60
      NNXI=XMIN
      MM=12
      MMS=1
      MMI=1
      DO 13 X=NNXI,NNX
      DO 13 S=MMI,MM,MMS
      W(S,X)=SQRT(W(S,X))
   13 U(S,X)=U(S,X)/N(S,X)
  603 FORMAT(' ',1P20E10.2)
      WRITE(7,604)
  604 FORMAT('0   TE')
      DO 2 X=NNXI,NNX
    2 WRITE(7,603) (TE(S,X),S=MMI,MM,MMS)
      WRITE(7,605)
  605 FORMAT('0   TI')
      DO 3 X=NNXI,NNX
    3 WRITE(7,603) (TI(S,X),S=MMI,MM,MMS)
      WRITE(7,606)
  606 FORMAT('0   N')
      DO 4 X=NNXI,NNX
    4 WRITE(7,603) (N(S,X),S=MMI,MM,MMS)
      WRITE(7,607)
  607 FORMAT('0   V')
      DO 5 X=NNXI,NNX
    5 WRITE(7,603) (V(S,X),S=MMI,MM,MMS)
      WRITE(7,608)
  608 FORMAT('0   VP')
      DO 6 X=NNXI,NNX
    6 WRITE(7,603) (W(S,X),S=MMI,MM,MMS)
```

```
      WRITE(7,609)
  609 FORMAT('0   E BOUNDARIES')
      DO 8 I=NNXI,NNX
    8 WRITE(7,603) (V(S,I),S=NNI,NN,NNS)
      IF(NTIME.LT.NRCH) GO TO 101
      GO TO 103
  102 WRITE(6,699) NTIME
  699 FORMAT('- LAST RECORD ON TAPE HAS NTIME=',I4,'.')
  103 RETURN
      END
```

```
C*********************************************************************
C    *    PLOT BEAM INTENSITIES VS. X AT VARIOUS TIMES.               *
C    *    BEAMS ARE BEAT FREQUENCY MIXED & COLLISIONALLY DAMPED.      *
C    *       RICHARD D. MILROY    SEPT. 13, 1977.                     *
C*********************************************************************
      REAL TE(1000),TI(1000),N(1000),L1(1000),L2(1000),I1(1000)
     1,I2(1000),K1(1000),K2(1000),K3(1000),K4(1000),LLMDA,L0,LP,LT,N0
     2,K10,L10,L20,I10,I20,I01(1000),I02(1000)
      N0=4.0E+16
      L10=10.6
      L20=10.615
      TE0=100.0
      TI0=100.0
      I10=1.0E+11
      I20=1.0E+10
      NP=900
      NPLTS=3
      L0=100.0
      LP=20.0
      LT=L0+LP
      DX=LT/(NP-1)
      NP1=NP*(L0/LT)
      NP1P1=NP1+1
      NPM1=NP-1
C------- FIND K1.
      LLMDA=ALOG(AMIN1(2.3*TE0**1.5,12.0*TE0))
      K10=8.67E-38*N0**2*L10**2*LLMDA/TE0**1.5
C------- INITIALIZE SOME ARRAYS.
      DO 1 I=1,NP
      TE(I)=TE0
      TI(I)=TI0
      N(I)=N0
      L1(I)=L10
      L2(I)=L20
    1 CONTINUE
      DO 2 I=1,NP1
      X=(NP1P1-I)*DX
      K1(I)=K10
      K3(I)=K10*(L20/L10)**2
      I1(I)=I10*EXP(-K10*X)
      I2(I)=1.0
    2 CONTINUE
      DO 3 I=NP1P1,NP
      K1(I)=0.0
      K2(I)=0.0
      K3(I)=0.0
      K4(I)=0.0
      I1(I)=I10
      I2(I)=1.0
    3 CONTINUE
      I2(1)=I20
      DO 9 I=1,NP
      I01(I)=I1(I)
    9 I02(I)=I2(I)
C------- FIND K2 AND K4.
      CALL BPMR(L1,L2,N,TE,TI,2,K2)
      DO 4 I=1,NP1
      K2(I)=K2(1)
      K4(I)=K2(1)*(L10/L20)
```

```
    4 CONTINUE
C------- INITIALIZE PLOTTING AND DRAW AXIS.
      CALL PLINT(LT,DX,NP)
      NTOTAL=NP-2
      NPLT=(NTOTAL/NPLTS)-1
C------- CALCULATE INTENSITIES.
      DO 5 J=1,NTOTAL
      DO 6 I=1,NPM1
      Z=0.5*(K1(I)+K2(I)*I02(I))*DX
    6 I1(I)=I01(I+1)*(1-Z)/(1+Z)
      DO 7 I=2,NP
      Z=0.5*(K3(I)-K4(I)*I01(I))*DX
    7 I2(I)=I02(I-1)*(1-Z)/(1+Z)
      DO 8 I=1,NP
      I01(I)=I1(I)
    8 I02(I)=I2(I)
C------- PLOT IF DESIRED.
      IF(((J-1)/NPLT)*NPLT.EQ.(J-1)) CALL PLT(I1,I2)
    5 CONTINUE
      CALL PLOT(0.0,0.0,999)
      STOP
      END
      SUBROUTINE PLINT(LT,DX,NP)
      [illegible],I1(1000),I2(1000),X(1000),LIMIN,LIMAX,LI1(1000),LI2(1000)
      [illegible]ONESEC(2)/0,1000000/
      [illegible]
      LIMIN=9.0
      LIMAX=14.0
      CALL PLOTS
      CALL ERASE(61)
      CALL TWAIT(0,ONESEC)
      CALL PLOT(20.0,10.0,-3)
      F=AMIN1(30.0/XLEN,20.0/YLEN)
      CALL FACTOR(F)
      GO TO 2
    1 CALL PLOT(3.0,3.0,-3)
    2 CONTINUE
C------- DRAW THE AXIS.
      DELTLG=LIMIN-LIMAX
      CALL LOGAX(0.0,0.0,'I',1,YLEN,90.0,LIMIN,DELTLG,0.0)
      AXDELT=LT/XLEN
      CALL AXIS2(0.0,0.0,'X',-1,XLEN,0.0,0.0,AXDELT,1.0)
      CALL PLOT(0.0,YLEN,3)
      CALL PLOT(XLEN,YLEN,2)
      CALL PLOT(XLEN,0.0,2)
C------- SET UP X - VECTOR.
      DO 3 I=1,NP
    3 X(I)=(I-1)*DX
      GO TO 999
      ENTRY PLT(I1,I2)
C------- TAKE LOG OF INTENSITIES.
      DO 4 I=1,NP
      LI1(I)=ALOG10(I1(I))
      LI2(I)=ALOG10(I2(I))
      IF(LI1(I).LT.LIMIN) LI1(I)=LIMIN
      IF(LI2(I).LT.LIMIN) LI2(I)=LIMIN
      IF(LI1(I).GT.LIMAX) LI1(I)=LIMAX
      IF(LI2(I).GT.LIMAX) LI2(I)=LIMAX
    4 CONTINUE
```

```
      LX1(NP+1) = LXMIN
      LX2(NP+1) = LXMIN
      LX1(NP+2) = (LXMAX-LXMIN)/YLEN
      LX2(NP+2) = LX1(NP+2)
      X(NP+1) = 0.0
      X(NP+2) = LT/XLEN
      CALL LINE(X,LX2,NP,1,0,0)
      CALL LINE(X,LX1,NP,1,0,0)
  999 RETURN
      END
      SUBROUTINE BFMR(L1,L2,N,TE,TI,NP,ALPHA)
C*********************************************************************
C     *   CALCULATE BEAT FREQUENCY MIXING RATES.                      *
C     *   BOTH ELECTRON & ION MOTION IS ACCOUNTED FOR.                *
C     *   ELECTRON - ION COLLISIONS ARE FULLY ACCOUNTED FOR.          *
C     *   PONDEROMOTIVE FORCE ON BOTH ELECTRONS AND IONS.             *
C     *   RICHARD D. MILROY        AUGUST, 10, 1977.                  *
C*********************************************************************
      COMPLEX J/(0.0,1.0)/,KE(1000),KI(1000),EP(1000),EDE(1000)
     1,EDI(1000),ECE(1000),ECI(1000),JE(1000),JI(1000),X,B(1000)
      REAL L1(1000),L2(1000),N(1000),TE(1000),TI(1000),W3(1000),K3(1000)
     1,W1(1000),W2(1000),ALPHA(1000),K1,K2
      C=3.[illegible]10
C------- [illegible] W3 AND K3.
      DO 1 I=1,NP
      KK=I
      W1(I)=1.885E+15/(L1(I))
      W2(I)=1.885E+15/(L2(I))
      W3(I)=W1(I)-W2(I)
      WPE=5.63E+4*SQRT(N(I))
      IF(WPE.GE.W2(I)) GO TO 8
      K1=SQRT(W1(I)**2-WPE**2)/C
      K2=SQRT(W2(I)**2-WPE**2)/C
    1 K3(I)=K1+K2
      GO TO 9
    8 NP=KK-1
C------- FIND LINEAR SUSCEPTABILITIES.
    9 CALL LSUS(W3,K3,N,TE,TI,NP,KE,KI)
C------- CALCULATE COLLISIONAL CORRECTION FACTOR.
      DO 7 I=1,NP
      VEI=2.53E-05*N(I)/(TE(I)**1.5)
      WPE=5.63E+4*SQRT(N(I))
      B(I)=1.0-J*W3(I)*VEI/(WPE**2)
      KE(I)=B(I)*KE(I)
      KI(I)=B(I)*KI(I)
    7 CONTINUE
C------- SPECIFY DIELECTRIC CONSTANT EP.
      DO 2 I=1,NP
    2 EP(I)=1.0+KE(I)+KI(I)
C------- SPECIFY PONDEROMOTIVE 'FIELDS'.    FOR I1=I2=1.0
      DO 3 I=1,NP
      X=K3(I)*L1(I)*L2(I)*J
      EDE(I)=3.11E-16*X
    3 EDI(I)=-1.69E-19*X
C------- SPECIFY COULOMB FIELDS.
      DO 4 I=1,NP
      ECE(I)=-KE(I)*(EDE(I)*(1.0[illegible](I)*[illegible](I))
    4 ECI(I)=-KI(I)*(EDI(I)*(1.0[illegible]DI(I)[illegible](I))
C------- SPECIFY ELECTRON AND ION CURRENTS.
      DO 5 I=1,NP
```

```
      X=J*W3(I)/12.566
      JE(I)=X*[illegible](I)
    5 JI(I)=X*ECI(I)
C------- SPECIFY HEAT FREQ[illegible]ING RATES.
      DO 6 I=1,NP
      X=2.0E-7*R1(I)/W3(I)
      A31R = X*REAL(EDE(I)[illegible](JE(I)))
      A32R = X*REAL(ECI(I)[illegible](JE(I)))
      A31I = X*REAL(EDI([illegible]G(JI(I)))
      A32I = X*REAL(ECE(I)[illegible]JG(JI(I)))
      A3R = A31R+A32R
      A3I = A31I+A32I
      ALPHA(I) = A3R+A3I
C     WRITE(6,601) L2(I),A31R,A32R,A3R,A31I,A32I,A3I,ALPHA(I)
  601 FORMAT(' ',10E12.4)
    6 CONTINUE
      RETURN
      END
      SUBROUTINE LSUS(W3,K3,N,TE,TI,NP,KE,KI)
C*****************************************************************************
C     * CALCULATE LINEAR SUSCEPTABILITIES FOR BOTH ELECTRONS & PROTONS. *
C     *          RICHARD D. KILROY        AUGUST, 14, 1977.              *
C*****************************************************************************
      COMPLEX KE(1000),KI(1000),J/(0.0,1.0)/,F0
      REAL W3(1000),K3(1000),N(1000),TE(1000),TI(1000)
C------- FIND KE.
      DO 1 I=1,NP
      WP=5.63E+4*SQRT(N(I))
      V0=5.93E+7*SQRT(TE(I))
      A0=W3(I)/(K3(I)*V0)
      IF(A0.GT.50.0) GO TO 3
      KE(I)=2.0*(WP*A0/W3(I))**2*(1.0+J*A0*F0(A0))
      GO TO 1
    3 KE(I)=-(WP/W3(I))**2
    1 CONTINUE
C------- FIND KI.
      DO 2 I=1,NP
      WP=1.31E+3*SQRT(N(I))
      V0=1.38E+6*SQRT(TI(I))
      A0=W3(I)/(K3(I)*V0)
      IF(A0.GT.50.0) GO TO 4
      KI(I)=2.0*(WP*A0/W3(I))**2*(1.0+J*A0*F0(A0))
      GO TO 2
    4 KI(I)=-(WP/W3(I))**2
    2 CONTINUE
      RETURN
      END
      COMPLEX FUNCTION F0(A0)
C------- CALCULATE F0 AS DEFINED ON PAGE 178 OF STIX.
      COMPLEX J/(0.0,1.0)/
      REAL FOSPL(12,5)
      DATA FOSPL/0.2,0.5,0.8,1.0,1.2,1.4,1.7,2.0,2.5,3.0,3.5,4.0,
     1 .3895,0.7[illegible],1.064,1.076,1.015,0.913,0.7451,0.6027,0.4462,0.3565
     2,0.2992,0.2507,1.003,1.299,0.5478,-0.2165,-0.4162,-0.5635,-0.[illegible],
     3-0.4114,-0.2316,-0.1393,-0.0931,0.0,0.0,0.987,-3.4911,-0.3208,
     4-0.6701,-0.06606,0.1709,0.231,0.1287,0.5593E-1,0.03642,0.0,1.097
     5,-4.976 ,5.272,-0.5696,1.007,0.2633,0.06673,-0.06821,-0.0485,
     6-0.01301,-0.02428,0.0/
       A0E=A0
       IF(A0E.GT.10.)A0E=10.0
```

```
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      DO [illegible]
      IF([illegible](I,1)) GO TO 8
 7    CONTINUE
 8    I=J-1
      [illegible]
      [illegible](I,2)+[illegible](I,3)*[illegible]+[illegible](I,4)*[illegible]
      [illegible]
      GO TO 11
 12   [illegible]*(1.0+[illegible]/([illegible])+[illegible].72/([illegible])+1.[illegible]/([illegible])+[illegible]/([illegible]))
      GO TO 11
 90   [illegible]=2.0*J*([illegible]-[illegible].[illegible]*[illegible]**3+.[illegible]*[illegible]**[illegible]-.0[illegible])
      GO TO 11
 11   [illegible]=1.[illegible]*EXP(-[illegible])
      [illegible]
      RETURN
      END
```

```
C**************************************************************************
C    *     SIMULATE INTERACTION OF TWO ANTIPARALLEL BEAMS IN A PLASMA     *
C    *     WITH A DIFFERENCE FREQUENCY OF THE ORDER OF THE ION-ACOUSTIC   *
C    *     FREQUENCY.  COST MAY BE PROHIBITIVE FOR LARGER DIFFERENCE      *
C    *     FREQUENCIES.   MAIN CALCULATIONS IN SUBROUTINE PCOM.           *
C    *  DEVELOPED  BY RICHARD G. MILROY.  OCTOBER 24 1977.                *
C**************************************************************************
      COMPLEX *16 D(100),A1(100),A2(100),W(100),K1,K2,A1BND,CDT,CDT2
     1,CW3,CTMAX,J/(0.0D0,1.0D0)/,T,CKIB1,CKIB2
      COMPLEX CMPLX
      REAL N0,LD1,LD2,K3,KA1,KA2,KIB1,KIB2,LLMDA
C------- SPECIFY INDEPENDENT VARIABLES.
      N0=2.5E+17
      TE0=160.0
      TI0=120.0
      LD1=10.2470
      LD2=10.2605
      ZMAX=80.0
      WMAX=6.0E+11
      PLEN=3.0
      NP=99
      NINT=41
      NPW=900
      NDT=10
      NPRT=30
      NTAP=5
      NREST=0
      P1=1.8E+10
      P21 = 1.0
      P22 = 1.E10
      P23 = 1.0
      P24 = 1.0
      P25 = 1.0
      Z21 = 0.0
      Z22 = 1.5
      Z23 = 3.0
      Z24 = 4.0
      Z25 = 5.0
      C=3.0E+10
C------- CALCULATE SOME DEPENDENT VARIABLES.
      W3=1.885E+15*(1./LD1-1./LD2)
      DZ=PLEN/(NP-1)
      W1=1.885E+15/LD1
      W2=1.885E+15/LD2
      WP=5.63E+04*SQRT(N0)
      C1 = C*SQRT(1.0-(WP/W1)**2)
      DT = DZ/(2.0*C1)
      DT2 = 2.0*DT
      TMAX = ZMAX/C1
      KA1 = SQRT(W1**2-WP**2)/C
      KA2 = SQRT(W1**2-WP**2)/C
      K3 = KA1 + KA2
      T=(0.0D0,0.0D0)
      NTIME=0
C------- CALCULATE INVERSE BREMSSTRAHLUNG COEFFICIENTS.
      LLMDA = ALOG(12.0*TE0)
      IF(TE0.LT.27.0) LLMDA = ALOG(2.3*TE0**1.5)
      GAM = 2.9E-06*N0*LLMDA/(TE0**1.5)
      KIB1 = GAM*WP**2/(2.0*W1**2)
```

```
      KIB2 = GAM*NP**2/(2.0*N2**2)
C------- PREPARE TO CALL PCOM  -  THE MAIN SUBROUTINE
      CALL DST(D,NINT,TMAX,K3,NN2,ZZ0,TI0,N0,NPN)
      X1 = SQRT(6.283E7/(W1*KA1))
      X2 = SQRT(6.283E7/(W2*KA2))
      DO 1 I=1,NP
      N(I)=(0.0D0,0.0D0)
      A1(I) = X1*SQRT(P1)*EXP(-KIB1*(TMAX))
    1 CONTINUE
C------- SPECIFY INITIAL PROFILE FOR PULSE.
      DO 2 I=1,NP
      Z = (I-1)*DZ
      IF(Z.GT.Z22) GO TO 3
      A2(I) = P21 + ((Z-Z21)/(Z22-Z21))*(P22-P21)
      GO TO 6
    3 CONTINUE
      IF(Z.GT.Z23) GO TO 4
      A2(I) = P22 + ((Z-Z22)/(Z23-Z22))*(P23-P22)
      GO TO 6
    4 CONTINUE
      IF(Z.GT.Z24) GO TO 5
      A2(I) = P23 + ((Z-Z23)/(Z24-Z23))*(P24-P23)
      GO TO 6
    5 CONTINUE
      IF(Z.GT.Z25) STOP
      A2(I) = P24 + ((Z-Z24)/(Z25-Z24))*(P25-P24)
      GO TO 6
6     A2(I) = X2*CDSQRT(A2(I))
    2 CONTINUE
      IF(NREST.NE.0)
     1CALL REST(NREST,NTIME,T,A1,A2,N)
      A1BND = X1*SQRT(P1)
      K1=-J*NP**2/(2.0*W1*N0)
      K2=W1*K1/W2
      CW3=CMPLX(W3,0.0)
      CTMAX=CMPLX(TMAX,0.0)
      CDT=CMPLX(DT,0.0)
      CDT2=CMPLX(DT2,0.0)
      CKIB1 = CMPLX(KIB1,0.0)
      CKIB2 = CMPLX(KIB2,0.0)
      CALL PCOM(A1,A2,N,D,CDT,CDT2,NDT,NINT,NP,K1,K2,A1BND,CW3,CTMAX
     1,W1,W2,NPRT,NTAP,T,NTIME,KA1,KA2,CKIB1,CKIB2)
      RETURN
      END
      SUBROUTINE PCOM(A1,A2,N,D,DT,DT2,NDT,NINT,NP,K1,K2,ABND
     1,W3,TMAX,W1,W2,NPRT,NTAP,T,NTIME,KA1,KA2,KIB1,KIB2)
C*********************************************************************
C     *          CALCULATE BEAM COMPRESSION DUE TO TWO ANTIPARALLEL     *
C     *          ELECTROMAGNETIC BEAMS WITH A DIFFERENCE FREQUENCY      *
C     *          CLOSE TO THE NATURAL ION-ACOUSTIC FREQUENCY.           *
C     *          TRANSIENT BEHAVIOUR IS INCLUDED.                       *
C     *          RICHARD D. MILROY   OCTOBER 23, 1977.                  *
C*********************************************************************
      IMPLICIT COMPLEX*16 (A-H,O-Z)
      COMPLEX*16 A1(100),A2(100),N(100),D(100),K1,K2
     1,A10(100),F(100),IJ/(0.0D0,1.0D0)/,KIB1,KIB2
      REAL W1,W2,KA1,KA2
      NDT1=NDT
      NDT=1
      DO 9 I=1,NP
```

```
    9 A10(I)=A1(I)
    1 CONTINUE
      A1END = CDEXP(-KI[illegible]*(TMAX-T))*AEND
      NTIME=NTIME+1
      IF(NTIME.GT.NP*2) NDT=NDT1
      DTN=DT*NDT
      T=T+DTN
C------- ADVANCE A1 BY DT - NDT TIMES.
      A1(1) = A1END
      DO 3 J=1,NDT
      A1IM1=A1END
      DO 2 I=2,NP
      IM1=I-1
      A1I=A1(I)
      A1(I)=A1IM1+DT*K1*(W(IM1)*A2(IM1)+W(I)*A2(I))-DT*KIB1*A1IM1
      A1IM1=A1I
    2 CONTINUE
    3 CONTINUE
C------- ADVANCE A2 BY DTN.
      NN = MIN0((NTIME+1)/2,NP)
      DO 4 I=1,NN
      A2(I)=A2(I)+DTN*K2*DCONJG(W(I))*A10(I)-DTN*KIB2*A2(I)
    4 CONTINUE
C------- MAKE OLD A1 EQUAL TO NEW A1.
      DO 5 I=1,NP
      A10(I)=A1(I)
    5 CONTINUE
C------- RECALCULATE W(I).
      NN=MIN0(NTIME/2,NP)
      IF(NN.LT.2) GO TO 8
      DO 7 I=2,NN
      NNN=MIN0(I,NINT)
      DO 6 J=1,NNN
      F(J)=A1(I-J+1)*DCONJG(A2(I-J+1))*D(J)*CDEXP(IJ*W3*(J-1)*DT2)*.3989
    6 CONTINUE
      CALL SIMP(ANS,F,NNN,DT2)
      W(I)=ANS
    7 CONTINUE
    8 CONTINUE
      IF((NTIME.EQ.1).OR.(((NTIME/NTAP)*NTAP).EQ.NTIME))
     1WRITE(3) T,NTIME,NP,DT2,A1,A2,W,W1,W2,KA1,KA2
      IF((NTIME.EQ.1).OR.(((NTIME/NPRT)*NPRT).EQ.NTIME))
     1CALL RITE(A1,A2,W,NTIME,NP,T,W1,W2,KA1,KA2)
      IF(CDABS(T).LT.CDABS(TMAX)) GO TO 1
      CALL RITE(A1,A2,W,NTIME,NP,T,W1,W2,KA1,KA2)
      RETURN
      END
      SUBROUTINE DET(D,NINT,WMAX,K3,DT,TR,TI,N,NPW)
C*************************************************************************
C     *     CALCULATE D;      D(K,T).                                    *
C     *     RICHARD D. MILROY   OCTOBER 24, 1977.                        *
C*************************************************************************
      COMPLEX*16 D(100),ANS
      COMPLEX DW(1000),F(1000),J/(0.0,1.0)/
      REAL W3(1000),T(100),K3,N
C------- CALCULATE DW;     D(K,W).
      DELW=2.0*WMAX/(NPW-1)
      DO 1 I=1,NPW
      W3(I)=-WMAX+DELW*(I-1)
      IF(ABS(W3(I)).LT.1.0E+08) W3(I)=1.0E+08
```

```
    1 CONTINUE
      CALL DEN(NPW,W3,K3,TE,TI,N,DW)
C------- CALCULATE D;      D(K,T).
      DO 2 JJ=1,NINT
      TT=DT*(JJ-1)
      T(JJ)=TT
      DO 3 I=1,NPW
      WT=W3(I)*TT
      F(I)=0.3989*DW(I)*CEXP(-J*WT)
    3 CONTINUE
      CALL SIMP1(ANS,F,NPW,DELW)
      D(JJ)=ANS
    2 CONTINUE
      D(1)=(0.0D0,0.0D0)
C------- MAKE PLOTS OF DW AND D IF DESIRED.
      CALL PLT(W3,T,DW,D,NPW,NINT)
      RETURN
      END
      SUBROUTINE DEN(NP,W3,K3,TE,TI,N,DW)
C*********************************************************************
C   *    CALCULATE DW;    D(K,W)                                      *
C   *         RICHARD D. MILROY     OCTOBER 24, 1977.                 *
C*********************************************************************
      COMPLEX DW(1000),KE(1000),KI(1000),B,J/(0.0,1.0)/,C
      REAL W3(1000),N,K3
C------- FIND LINEAR SUSCEPTABILITIES.
      CALL LSUS(W3,K3,N,TE,TI,NP,KE,KI)
C------- CALCULATE DW.
      VEI=2.53E-05*N/(TE**1.5)
      WPE=5.63E+04*SQRT(N)
      C=-K3**2*9.716E+04
      DO 1 I=1,NP
      B=1.0-J*W3(I)*VEI/(WPE**2)
      DW(I)=C*KE(I)*(1.0+B*KI(I))/(1.0+B*KE(I)+B*KI(I))
    1 CONTINUE
      RETURN
      END
      SUBROUTINE LSUS(W3,K3,N,TE,TI,NP,KE,KI)
C*********************************************************************
C    * CALCULATE LINEAR SUSCEPTABILITIES FOR BOTH ELECTRONS & PROTONS. *
C    *          RICHARD D. MILROY      AUGUST, 14, 1977.              *
C*********************************************************************
      COMPLEX KE(1000),KI(1000),J/(0.0,1.0)/,FO
      REAL W3(1000),K3,N
C------- FIND KE.
      WP=5.63E+4*SQRT(N)
      VO=5.93E+7*SQRT(TE)
      DO 1 I=1,NP
      AO=W3(I)/(K3*VO)
      IF(AO.GT.50.0) GO TO 3
      KE(I)=2.0*(WP*AO/W3(I))**2*(1.0+J*AO*FO(AO))
      GO TO 1
    3 KE(I)=-(WP/W3(I))**2
    1 CONTINUE
C------- FIND KI.
      WP=1.31E+3*SQRT(N)
      VO=1.38E+6*SQRT(TI)
      DO 2 I=1,NP
      AO=W3(I)/(K3*VO)
      IF(AO.GT.50.0) GO TO 4
```

```
      KI(I)=2.0*(WP*A0/W3(I))**2*(1.0+J*A0*F0(A0))
      GO TO 2
    4 KI(I)=-(WP/W3(I))**2
    2 CONTINUE
      RETURN
      END
      COMPLEX FUNCTION F0(AA0)
C------- CALCULATE F0 AS DEFINED ON PAGE 178 OF STIX.
      COMPLEX J/(0.0,1.0)/
      REAL F0SPL(12,5)
      DATA F0SPL/0.2,0.5,0.8,1.0,1.2,1.4,1.7,2.0,2.5,3.0,3.5,4.0,
     1 .3695,0.7199,1.064,1.076,1.015,0.913,0.7451,0.6027,0.4462,0.3565
     2,0.2992,0.2587,1.003,1.299,0.5474,-0.2165,-0.4162,-0.5635,-0.532,
     3-0.4114,-0.2316,-0.1393,-0.0931,0.0,0.0,0.987,-3.4911,-0.3284,
     4-0.6701,-0.06606,0.1709,0.231,0.1287,0.5593E-1,0.03642,0.0,1.097
     5,-4.976 ,5.272,-0.5696,1.007,0.2633,0.06673,-0.06821,-0.0485,
     6-0.01301,-0.02428,0.0/
      A0 = ABS(AA0)
      A0E=A0
      IF(A0E.GT.10.)A0E=10.0
      A0ES=A0E*A0E
C     SPLINE APPROXIMATION FOR F0
      IF(A0.LT.0.21) GO TO 90
      IF(A0.GT.3.25)GO TO 12
      DO7 I=1,12
      IF(A0.LT.F0SPL(I,1))GO TO 8
    7 CONTINUE
    8 I=I-1
      DA=A0-F0SPL(I,1)
      F0=F0SPL(I,2)+F0SPL(I,3)*DA+F0SPL(I,4)*DA*DA+F0SPL(I,5)*DA**3
      F0=J*F0
      GO TO 11
   12 F0=J/A0*(1.0+0.5/(A0*A0)+0.75/(A0**4)+1.875/(A0**6)+6.563/(A0**8))
      GO TO 11
   90 F0=2.0*J*(A0-0.666667*A0**3+.2666667*A0**5-.0761905*A0**7)
      GO TO 11
   11 DAMP=1.772*EXP(-A0ES)
      F0 = (AA0/A0)*F0
      F0=F0+DAMP
      RETURN
      END
      SUBROUTINE SIMP(ANS,F,N,DT)
C***********************************************************************
C    *      INTEGRATE F USING SIMPSON'S RULE.                           *
C    *      IF N IS NOT ODD INTEGRATE TO N-1 AND ADD REMAINDER.         *
C    *        RICHARD D. MILROY OCTOBER 23, 1977.                       *
C***********************************************************************
      IMPLICIT COMPLEX*16(A-H,O-Z)
      COMPLEX*16 F(100)
      IF(N.EQ.2) GO TO 1
      GO TO 2
    1 ANS=(0.5D0,0.0D0)*DT*(F(1)+F(2))
      GO TO 6
    2 NTEST=1
      IF(((N/2)*2).EQ.N) NTEST=2
      N1=N
      IF(NTEST.EQ.2) N1=N-1
      NN1=N1-1
      NN2=N1-2
      F1=F(1)+F(N1)
```

```
      F2=(0.0D0,0.0D0)
      F3=(0.0D0,0.0D0)
      DO 3 I=2,NN1,2
    3 F2=F2+F(I)
      DO 4 I=3,NN2,2
    4 F3=F3+F(I)
      ANS=(DT/(3.0D0,0.0D0))*(F1+(4.0D0,0.0D0)*F2+(2.0D0,0.0D0)*F3)
      GO TO (6,5),NTEST
    5 ANS=ANS+(F(N1)+F(N))*DT*(0.5D0,0.0D0)
    6 RETURN
      END
      SUBROUTINE SIMP1(ANS,F,N,DT)
C*********************************************************************
C     *     INTEGRATE F USING SIMPSON'S RULE.                        *
C     *     IF N IS NOT ODD INTEGRATE TO N-1 AND ADD REMAINDER.      *
C     *       RICHARD D. MILROY OCTOBER 23, 1977.                    *
C*********************************************************************
      COMPLEX*16 ANS
      COMPLEX F(1000),F1,F2,F3
      IF(N.EQ.2) GO TO 1
      GO TO 2
    1 ANS=(0.5,0.0)*DT*(F(1)+F(2))
      GO TO 6
    2 NTEST=1
      IF(((N/2)*2).EQ.N) NTEST=2
      N1=N
      IF(NTEST.EQ.2) N1=N-1
      NN1=N1-1
      NN2=N1-2
      F1=F(1)+F(N1)
      F2=(0.0,0.0)
      F3=(0.0,0.0)
      DO 3 I=2,NN1,2
    3 F2=F2+F(I)
      DO 4 I=3,NN2,2
    4 F3=F3+F(I)
      ANS=(DT/(3.0,0.0))*(F1+(4.0,0.0)*F2+(2.0,0.0)*F3)
      GO TO (6,5),NTEST
    5 ANS=ANS+(F(N1)+F(N))*DT*(0.5,0.0)
    6 RETURN
      END
      SUBROUTINE REST(NREST,NTIME,T,A1,A2,N)
C*********************************************************************
C     *     READ DATA FROM DISK TO ALLOW ROUTINE TO RESTART.         *
C     *           RICHARD D. MILROY     OCTOBER 26, 1977.            *
C*********************************************************************
      COMPLEX*16 A1(100),A2(100),N(100),T,DT2
    1 READ(2,END=999) T,NTIME
      IF(NTIME.NE.NREST) GO TO 1
      BACKSPACE 2
      READ(2) T,NTIME,NP,DT2,A1,A2,N,N1,N2
      RETURN
  999 WRITE(6,601) NREST
  601 FORMAT('1','RECORD NREST=',I5,'  IS NOT ON UNIT 2.')
      STOP
      END
      SUBROUTINE PLT(W3,T,DW,DT,NPW,NPT)
      COMPLEX*16 DT(100)
      COMPLEX DW(1000),ZZZ
      REAL W3(1000),T(100),RW(1000),IW(1000),IT(1000),F(2000)
```

```
      [illegible]ESEC(2)/0, 1000000/
      [illegible]
      [illegible].0
      XINT1 = 5.0
      YINT = 6.0
      CALL PLOTS
      CALL ORG1(XLEN,YLEN)
      CALL TERASE(&1)
      CALL TWAIT(0,ONESEC)
    1 CONTINUE
      DO 2 I=1,NPW
      RW(I) = REAL(DW(I))
    2 IW(I) = AIMAG(DW(I))
      DO 3 I= 1,NPT
      ZZZ=DT(I)
    3 IT(I) = REAL(ZZZ)
C------- SCALE QUANTITIES TO BE PLOTTED.
      DO 4 I=1,NPW
      F(I) = RW(I)
    4 F(I+NPW) = IW(I)
      NPW2 = 2*NPW
      CALL SCALE(W3,XINT,NPW,1)
      CALL SCALE(T,XINT1,NPT,1)
      CALL SCALE(F,YINT,NPW2,1)
      CALL SCALE(IT,YINT,NPT,1)
      W3(NPW+2) = W3(NPW+2)*XINT/XLEN
      T(NPT+2) = T(NPT+2)*XINT1/XLEN
      F(NPW2+2) = F(NPW2+2)*YINT/YLEN
      IT(NPT+2) = IT(NPT+2)*YINT/YLEN
      RW(NPW+1) = F(NPW2+1)
      IW(NPW+1) = F(NPW2+1)
      RW(NPW+2) = F(NPW2+2)
      IW(NPW+2) = F(NPW2+2)
C-------- DRAW THE AXIS FOR FREQUENCY SPECTRUM PLOT.
      AXMIN = W3(NPW+1)
      AXDELT = W3(NPW+2)
      DTIC = XLEN/XINT
      CALL AXIS2(0.0,0.0,'W',-1,XLEN,0.0,AXMIN,AXDELT,DTIC)
      CALL PLOT(0.0,YLEN,3)
      CALL PLOT(XLEN,YLEN,2)
      CALL PLOT(XLEN,0.0,2)
      AXMIN = RW(NPW+1)
      AXDELT = RW(NPW+2)
      DTIC = -YLEN/XINT
      CALL AXIS2(0.0,0.0,'D',1,YLEN,90.0,AXMIN,AXDELT,DTIC)
C------- PLOTL THE PLOT.
      CALL LINE(W3,RW,NPW,1,0,0)
      CALL LINE(W3,IW,NPW,1,0,0)
      CALL ORG(XLEN,YLEN)
C------- DRAW AXIS FOR T PLOT.
      AXMIN = T(NPT+1)
      AXDELT = T(NPT+2)
      DTIC = XLEN/XINT1
      CALL AXIS2(0.0,0.0,'T',-1,XLEN,0.0,AXMIN,AXDELT,DTIC)
      CALL PLOT(0.0,YLEN,3)
      CALL PLOT(XLEN,YLEN,2)
      CALL PLOT(XLEN,0.0,2)
      AXMIN = IT(NPT+1)
      AXDELT = IT(NPT+2)
```

```
      DTIC = -YLEN/XINT
      CALL AXIS2(0.0,0.0,'D',1,YLEN,90.0,AXMIN,AXDELT,DTIC)
C------- PLOT THE PLOT.
      CALL LINE(T,IT,NPT,1,0,0)
      CALL PLOT(0.0,0.0,999)
      RETURN
      END
      SUBROUTINE ORG1(XLEN,YLEN)
C---------MOVE ORIGIN IN A WAY TO MINIMIZE PLOTTER PAPER WASTE.
C---------FOR TEKTRONIX, RE-SCALE PLOT AND HALT UNTILL RETURN ENTERED.
      CALL TCLEAR(&1)
      CALL TERASE(&1)
      P=AMIN1(30.0/XLEN,20[illegible])
      CALL FACTOR(P)
      CALL PLOT(2.0,2.0,-3)
      GO TO 10
    1 CALL PLOT(3.0,3.0,-3)
      X0=0.0
      Y0=0.0
      XL=XLEN
      YL=YLEN
      GO TO 10
      ENTRY ORG(XLEN,YLEN)
C---------ENTER HERE FOR ALL BUT 1'ST CALL TO THIS ROUTINE.
      CALL TCLEAR(&2)
      READ(6,601) NOTNG
  601 FORMAT(I5)
      CALL TERASE(&2)
      GO TO 10
    2 VX=6.0+XL
      VY=6.0+YL
      Y=Y0+VY+YLEN
      YL=YLEN
      IF(Y.GT.25.0) GO TO 3
      CALL PLOT(0.0,VY,-3)
      Y0=Y0+VY
      IF(XLEN.GT.XL) XL=XLEN
      GO TO 10
    3 VY=-Y0
      CALL PLOT(VX,VY,-3)
      XL=XLEN
      X0=X0+VX
      Y0=0.0
   10 RETURN
      END
      SUBROUTINE RITE(A1,A2,N,NTIME,NP,TIME,W1,W2,KA1,KA2)
      COMPLEX*16 A1(100),A2(100),N(100),TIME
      COMPLEX T1
      REAL P1,P2,T,KA1,KA2
      C=3.0E+10
      T1=TIME
      T=REAL(T1)
      WRITE(6,601) NTIME,T
  601 FORMAT('0','NTIME=',I5,'    T=',1PE10.2,' SEC.')
      NP1=NP
      NSKIP=10
      DO 1 I=1,NP1,NSKIP
      P1 = W1*KA1*(CDABS(A1(I)))**2/6.283E+07
      P2 = W2*KA2*(CDABS(A2(I)))**2/6.283E+07
      XN=CDABS(N(I))
```

```
      WRITE(6,602) I,P1,P2,XN
    1 CONTINUE
  602 FORMAT(' ',I5,1P3E10.2)
      RETURN
      END
```

```
C*****************************************************************************
C     *             PLOT THE OUTPUT FROM PCOM  -  PULSE COMPRESSION.          *
C     *                RICHARD D. MILROY   OCTOBER 26, 1977.                  *
C*****************************************************************************
      COMPLEX*16 T,DT2,A1(100),A2(100),N(100)
      COMPLEX Z
      REAL P1(102),P2(102),N1(102),X(102),KA1,KA2
      XLEN = 3.5
      YLEN = 3.5
      XINT = 6.0
      LPMIN = 7
      LPMAX = 13
      LNMIN = 12
      LNMAX = 17
      C = 3.0E+10
      READ(3) T,NTIME,NP,DT2,A1,A2,N,W1,W2,KA1,KA2
      BACKSPACE 3
      C1 = KA1/W1*C**2
      CALL PLOTS
      CALL ORG1(XLEN,YLEN)
      ALX=0.0
      ALY=800
      TR=-10.0
    1 CONTINUE
      CALL SETAL(ALX,ALY,1,&100,&100)
  100 WRITE(6,601)
  601 FORMAT('OINPUT Z')
      READ(5,501) DIST
      IF(DIST.LT.0.0) GO TO 999
  501 FORMAT(E10.3)
      TPLT = DIST/C1
      IF(TPLT.LT.TR) REWIND 3
C------- READ IN VALUE FROM DISK CLOSEST TO DESIRED PLOT TIME.
    2 TOLD=TR
      READ(3,END=8) T
      TR=CDABS(T)
      IF(TR.LT.TPLT) GO TO 2
      IF((TPLT-TOLD).GT.(TR-TPLT)) GO TO 7
    8 BACKSPACE 3
    7 BACKSPACE 3
      READ(3) T,NTIME,NP,DT2,A1,A2,N,W1,W2,KA1,KA2
C------- DEFINE X - VECTOR.
      TR=CDABS(T)
      DT=CDABS(DT2)
      DZ=C1*DT
      DO 3 I=1,NP
      X(I)=(I-1)*DZ
    3 CONTINUE
C------- DEFINE Y - VECTORS.
      DO 4 I=1,NP
      P1(I) = W1*KA1*(CDABS(A1(I)))**2/6.283E+07
      P2(I) = W2*KA2*(CDABS(A2(I)))**2/6.283E+07
      N1(I)=2.0*CDABS(N(I))  +  1.0
    4 CONTINUE
C------- TAKE LOG OF QUANTITIES TO BE PLOTTED.
      DO 5 I=1,NP
      P1(I)=ALOG10(P1(I))
      P2(I)=ALOG10(P2(I))
      N1(I)=ALOG10(N1(I))
```

```
    5 CONTINUE
C------- MAKE SURE ALL QUANTITIES STAY WITHIN GRAPH BOUNDARIES.
      DO 6 I=1,NP
      IF(P1(I).LT.LPMIN) P1(I)=LPMIN
      IF(P1(I).GT.LPMAX) P1(I)=LPMAX
      IF(P2(I).LT.LPMIN) P2(I)=LPMIN
      IF(P2(I).GT.LPMAX) P2(I)=LPMAX
      IF(M1(I).LT.LMMIN) M1(I)=LMMIN
      IF(M1(I).GT.LMMAX) M1(I)=LMMAX
    6 CONTINUE
C------- SCALE THE VECTORS TO BE PLOTTED.
      CALL SCALE(X,XINT,NP,1)
      X(NP+2) = X(NP+2)*XINT/XLEN
      P1(NP+1)=LPMIN
      P2(NP+1)=LPMIN
      M1(NP+1)=LMMIN
      P1(NP+2)=FLOAT(LPMAX-LPMIN)/YLEN
      P2(NP+2)=P1(NP+2)
      M1(NP+2)=FLOAT(LMMAX-LMMIN)/YLEN
C------- PLOT THE AXIS.
      AXMIN =X(NP+1)
      AXDELT=X(NP+2)
      DTIC = XLEN/XINT
      CALL AXIS2(0.0,0.0,'X (CM)',-6,XLEN,0.0,AXMIN,AXDELT,DTIC)
      CALL PLOT(0.0,YLEN,3)
      CALL PLOT(XLEN,YLEN,2)
      DELTLG=FLOAT(LPMIN-LPMAX)
      AXMINI=FLOAT(LPMIN)
      CALL LOGAX(0.0,0.0,'P (N/CM**2)',11,YLEN,90.0,AXMINI,DELTLG,3.0)
      DELTLG=FLOAT(LMMIN-LMMAX)
      AXMINI=FLOAT(LMMIN)
      CALL LOGAX(XLEN,0.0,'M',-1,YLEN,90.0,AXMINI,DELTLG,3.0)
C------- WRITE POSITION OF BEAMFRONT IN LOWER LEFT CORNER.
      DIST = C1*TE
      VX=0.0
      VY=-0.5
      HT=0.1
      CALL NUMBER(VX,VY,HT,DIST,0.0,1)
C------- DRAW THE LINES.
      CALL LINE(X,P1,NP,1,0,1)
      CALL LINE(X,P2,NP,1,0,1)
      CALL LINE(X,M1,NP,1,10,3)
C------- MOVE ORIGIN UNLESS ON TEKTRONIX.
      CALL ORG(XLEN,YLEN)
      GO TO 1
  999 CALL PLOT(0.0,0.0,999)
      STOP
      END
      SUBROUTINE ORG1(XLEN,YLEN)
C---------MOVE ORIGIN IN A WAY TO MINIMIZE PLOTTER PAPER WASTE.
C---------FOR TEKTRONIX, RE-SCALE PLOT AND HALT UNTILL RETURN ENTERED.
      INTEGER ONESEC(2)/0,1000000/
      CALL TCLEAR(61)
      CALL TERASE(61)
      CALL TWAIT(0,ONESEC)
      F=AMIN1(30.0/XLEN,20.0/YLEN)
      CALL FACTOR(F)
      CALL PLOT(3.0,2.0,-3)
      GO TO 10
    1 CALL PLOT(5.0,5.0,-3)
```

```
      X0=0.0
      Y0=0.0
      XL=XLEN
      YL=YLEN
      GO TO 10
      ENTRY ORG(XLEN,YLEN)
C---------ENTER HERE FOR ALL BUT 1'ST CALL TO THIS ROUTINE.
      CALL TCLEAR(62)
      READ(6,601) NOTNG
  601 FORMAT(I5)
      CALL TERASE(62)
      CALL TWAIT(0,ONESEC)
      GO TO 10
    2 VX=6.0+XL
      VY=6.0+YL
      Y=Y0+VY+YLEN
      YL=YLEN
      IF(Y.GT.20.0) GO TO 3
      CALL PLOT(0.0,VY,-3)
      Y0=Y0+VY
      IF(XLEN.GT.XL) XL=XLEN
      GO TO 10
    3 VY=-Y0
      CALL PLOT(VX,VY,-3)
      XL=XLEN
      X0=X0+VX
      Y0=0.0
   10 RETURN
      END
```

```
C*****************************************************************************
C    *      WILL PLOT ALPHA VS. ANY VARIABLE.   (DEPENDING ON 'CODE').        *
C    *      CODE = 0 STOP                                                     *
C    *           =+1 OR -1  PLOT ALPHA VS. TE.                                *
C    *           =+2 OR -2  PLOT ALPHA VS. TI.                                *
C    *           =+3 OR -3  PLOT ALPHA VS. N.                                 *
C    *           =+4 OR -4  PLOT ALPHA VS. L2.                                *
C    *           =+5 OR -5  PLOT ALPHA VS. TE.   (SEE TI=TE/N.)               *
C    *      IF (CODE.LT.0) NO NEW AXIS IS MADE.  PLOTS OVERLAY.               *
C    *           RICHARD D. KILROY        AUGUST, 15, 1977.                   *
C*****************************************************************************
      REAL L1(1000),L2(1000),N(1000),TE(1000),TI(1000),X(1000),Y(1000)
     1,ALPHA(1000),L2MIN,L2MAX,ND,L20,L10,ND,L2D,L2MINP,L2MAXP,W(1000)
     2,NE(1000),NI(1000)
      INTEGER CODE,C1,ONESEC(2)/0,1000000/
      NTIME=0
      NP=998
      NPI=NP
      XLEN=3.5
      YLEN=3.5
      LTEMIN=0
      LTEMAX=4
      LTIMIN=0
      LTIMAX=4
      LNMIN=16
      LNMAX=19
      LWMIN=10
      LWMAX=14
      L2MIN=10.6
      L2MAX=10.606
C------- L2MINP AND L2MAXP USED FOR X-AXIS LIMITS WHEN CODE=4.
      L2MINP=10.6
      L2MAXP=10.606
      LYMIN=-15
      LYMAX=-10
      LY2MIN=-16
      LY2MAX=-11
      L10=9.6
      ALX=0.0
      ALY=800.0
      TED=FLOAT(LTEMAX-LTEMIN)/NP
      TID=FLOAT(LTIMAX-LTIMIN)/NP
      ND=FLOAT(LNMAX-LNMIN)/NP
      L2D=(L2MAX-L2MIN)/NP
      CALL PLOTS
      CALL ORG1(XLEN,YLEN)
      CALL TERASE(64)
C------- PAUSE ONE SECOND WHILE TEKTRONIX SCREEN IS ERASED.
      CALL TWAIT(0,ONESEC)
    1 CONTINUE
      NP=NPI
      NTIME=NTIME+1
      ALY=ALY-60.0
      CALL SETAL(ALX,ALY,1,&100,&100)
  100 CONTINUE
      WRITE(6,601)
  601 FORMAT('0','INPUT CODE 1-TE; 2-TI; 3-N; 4-L2; 5-T; 6-N;')
      READ(5,501) CODE
  501 FORMAT(I3)
```

```
      IF(IABS(CODE).GT.0) GO TO 2
      CALL PLOT(0.0,0.0,999)
      STOP
    2 CONTINUE
      C1=IABS(CODE)
C-------- READ IN PARAMETERS.
      GO TO (11,12,13,14,15,14),C1
   11 WRITE(6,602)
  602 FORMAT('  INPUT TIO, NO L20')
      READ(5,502) TIO,NO,L20
  502 FORMAT(3E10.3)
      GO TO 16
   12 WRITE(6,603)
  603 FORMAT('  INPUT TEO, NO,, L20')
      READ(5,502) TEO,NO,L20
      GO TO 16
   13 WRITE(6,604)
  604 FORMAT('  INPUT TEO, TIO, L20')
      READ(5,502) TEO,TIO,L20
      GO TO 16
   14 WRITE(6,605)
  605 FORMAT('  INPUT TEO, TIO, NO')
      READ(5,502) TEO,TIO,NO
      GO TO 16
   15 WRITE(6,610)
  610 FORMAT('  INPUT NO, L20, TE/TI')
      READ(5,502) NO,L20,R
   16 CONTINUE
C------- IF(CODE.GT.0) ERASE SCREEN & RE-WRITE INITIAL CONDITIONS.
      IF(CODE.LT.0) GO TO 70
      CALL TERASE(670)
C------- PAUSE ONE SECOND WHILE TEKTRONIX SCREEN IS ERASED.
      CALL TWAIT(0,ONESEC)
      ALY=740
      CALL SETAL(ALX,ALY,1,670,670)
      WRITE(6,601)
      WRITE(6,606) CODE
  606 FORMAT(' ',I3)
      GO TO(61,62,63,64,65,64),C1
   61 WRITE(6,602)
      WRITE(6,607) TIO,NO,L20
  607 FORMAT(' ',F7.0,1PE9.1,0PF8.4)
      GO TO 70
   62 WRITE(6,603)
      WRITE(6,607) TEO,NO,L20
      GO TO 70
   63 WRITE(6,604)
      WRITE(6,608) TEO,TIO,L20
  608 FORMAT(' ',2F7.0,F8.4)
      GO TO 70
   64 WRITE(6,605)
      WRITE(6,609) TEO,TIO,NO
  609 FORMAT(' ',2F7.0,1PE9.1)
      GO TO 70
   65 WRITE(6,610)
      WRITE(6,611) NO,L20,R
  611 FORMAT(' ',1PE9.1,0PF8.4,F6.2)
   70 CONTINUE
C------- MOVE ORIGIN AND PLOT NEW AXIS IF DESIRED.
      IF(CODE.LT.0) GO TO 3
```

```
      IF(NTIME.NE.1) CALL ORG(XLEN,YLEN)
      CALL AXPL(C1,XLEN,YLEN,LYMIN,LYMAX,LTEMIN,LTEMAX,LTIMIN
     1,LTIMAX,LNMIN,LNMAX,L2MINP,L2MAXP,LWMIN,LWMAX
     2,LY2MIN,LY2MAX)
    3 CONTINUE
C-------- SPECIFY PLASMA PARAMETERS.
      IF((C1.EQ.1).OR.(C1.EQ.5)) GO TO 4
      DO 5 I=1,NP
    5 TE(I)=TE0
    4 CONTINUE
      IF((C1.EQ.2).OR.(C1.EQ.5)) GO TO 6
      DO 7 I=1,NP
    7 TI(I)=TI0
    6 CONTINUE
      IF(C1.EQ.3) GO TO 8
      DO 9 I=1,NP
    9 N(I)=N0
    8 CONTINUE
      IF((C1.EQ.4).OR.(C1.EQ.6)) GO TO 10
      DO 20 I=1,NP
   20 L2(I)=L20
   10 CONTINUE
      DO 21 I=1,NP
   21 L1(I)=L10
C-------- SPECIFY VARYING PARAMETER.
      GO TO (31,32,33,34,35,43),C1
   31 DO 36 I=1,NP
   36 TE(I)=10.0**(LTEMIN+TED*I)
      GO TO 40
   32 DO 37 I=1,NP
   37 TI(I)=10.0**(LTIMIN+TID*I)
      GO TO 40
   33 DO 38 I=1,NP
   38 N(I)=10.0**(LNMIN+ND*I)
      GO TO 40
   34 DO 39 I=1,NP
   39 L2(I)=L2MIN+L2D*I
      GO TO 40
   35 DO 42 I=1,NP
      TE(I)=10.0**(LTEMIN+TED*I)
   42 TI(I)=TE(I)/R
      GO TO 40
   43 DLW = FLOAT(LWMAX-LWMIN)/NP
      DO 44 I=1,NP
      W(I)=10.0**(LWMIN+DLW*I)
   44 L2(I) = 1.885E+15*L10/(1.885E+15-W(I)*L10)
   40 CONTINUE
C-------- CALCULATE MIXING RATE (ALPHA).
      CALL BFMR(L1,L2,N,TE,TI,NP,ALPHA,NE,NI)
C-------- SPECIFY Y-VECTOR FOR PLOTTING
      DO 41 I=1,NP
      Y(I)=ALOG10(ALPHA(I))
      IF(Y(I).GT.LYMAX) Y(I)=LYMAX
      IF(Y(I).LT.LYMIN) Y(I)=LYMIN
      NE(I)=ALOG10(NE(I))
      NI(I)=ALOG10(NI(I))
      IF(NE(I).GT.LY2MAX) NE(I)=LY2MAX
      IF(NE(I).LT.LY2MIN) NE(I)=LY2MIN
      IF(NI(I).GT.LY2MAX) NI(I)=LY2MAX
      IF(NI(I).LT.LY2MIN) NI(I)=LY2MIN
```

```
   41 CONTINUE
      Y(NP+1)=LYMIN
      Y(NP+2)=FLOAT(LYMAX-LYMIN)/YLEN
      NE(NP+1)=LY2MIN
      NI(NP+1)=LY2MIN
      NE(NP+2)=FLOAT(LY2MAX-LY2MIN)/YLEN
      NI(NP+2)=FLOAT(LY2MAX-LY2MIN)/YLEN
C-------- SPECIFY X-VECTOR FOR PLOTTING.
      GO TO (51,52,53,54,51,66),C1
   51 X(NP+1)=LTEMIN
      X(NP+2)=FLOAT(LTEMAX-LTEMIN)/XLEN
      DO 56 I=1,NP
   56 X(I)=ALOG10(TE(I))
      GO TO 60
   52 X(NP+1)=LTIMIN
      X(NP+2)=FLOAT(LTIMAX-LTIMIN)/XLEN
      DO 57 I=1,NP
   57 X(I)=ALOG10(TI(I))
      GO TO 60
   53 X(NP+1)=LNMIN
      X(NP+2)=FLOAT(LNMAX-LNMIN)/XLEN
      DO 58 I=1,NP
   58 X(I)=ALOG10(N(I))
      GO TO 60
   54 X(NP+1)=L2MINP
      X(NP+2)=(L2MAXP-L2MINP)/XLEN
      DO 59 I=1,NP
   59 X(I)=L2(I)
      GO TO 60
   66 X(NP+1)=LWMIN
      X(NP+2) = FLOAT(LWMAX-LWMIN)/XLEN
      DO 67 I=1,NP
   67 X(I) = ALOG10(W(I))
   60 CONTINUE
C-------- DRAW THE LINE.
      CALL LINE(X,Y,NP,1,0,0)
      ICHREP = NP/20
      CALL LINE(X,NE,NP,1,ICHREP,1)
      CALL LINE(X,NI,NP,1,ICHREP,3)
      CALL TCLEAR(61)
      GO TO 1
      END
      SUBROUTINE AXPL(C1,XLEN,YLEN,LYMIN,LYMAX,LTEMIN,LTEMAX
     1,LTIMIN,LTIMAX,LNMIN,LNMAX,L2MINP,L2MAXP,LWMIN,LWMAX
     2,LY2MIN,LY2MAX)
      REAL L2MINP,L2MAXP
      INTEGER C1
C-------- PLOT THE Y-AXIS.
      AXMIN=LYMIN
      DELTG=LYMIN-LYMAX
      CALL LOGAX(0.0,0.0,'ALPHA',5,YLEN,90.0,AXMIN,DELTG,3.0)
      AXMIN=LY2MIN
      DELTG=LY2MIN-LY2MAX
      CALL LOGAX(XLEN,0.0,'N',-1,YLEN,90.0,AXMIN,DELTG,3.0)
C-------- PLOT THE X-AXIS.
      GO TO (1,2,3,4,1,6),C1
    1 AXMIN=LTEMIN
      DELTG=LTEMIN-LTEMAX
      CALL LOGAX(0.0,0.0,'TE',-2,XLEN,0.0,AXMIN,DELTG,0.0)
      CALL LOGAX(0.0,YLEN,' ',1,XLEN,0.0,AXMIN,DELTG,0.0)
```

```
      GO TO 10
    2 AXMIN=LTIMIN
      DELTG=LTIMIN-LTIMAX
      CALL LOGAX(0.0,0.0,'TI',-2,XLEN,0.0,AXMIN,DELTG,0.0)
      CALL LOGAX(0.0,YLEN,' ',1,XLEN,0.0,AXMIN,DELTG,0.0)
      GO TO 10
    3 AXMIN=LNMIN
      DELTG=LNMIN-LNMAX
      CALL LOGAX(0.0,0.0,'N',-2,XLEN,0.0,AXMIN,DELTG,0.0)
      CALL LOGAX(0.0,YLEN,' ',1,XLEN,0.0,AXMIN,DELTG,0.0)
      GO TO 10
    4 AXMIN=L2MINP
      DELTG=(L2MAXP-L2MINP)/XLEN
      CALL AXIS2(0.0,0.0,'L2',-2,XLEN,0.0,AXMIN,DELTG,1.0)
      CALL AXIS2(0.0,YLEN,' ',1,XLEN,0.0,AXMIN,DELTG,1.0)
      GO TO 10
    6 AXMIN=LNMIN
      DELTG=LNMIN-LNMAX
      CALL LOGAX(0.0,0.0,'N',-1,XLEN,0.0,AXMIN,DELTG,0.0)
      CALL LOGAX(0.0,YLEN,' ',1,XLEN,0.0,AXMIN,DELTG,0.0)
   10 RETURN
      END
      SUBROUTINE ORG1(XLEN,YLEN)
C---------MOVE ORIGIN IN A WAY TO MINIMIZE PLOTTER PAPER WASTE.
C---------FOR TEKTRONIX, RE-SCALE PLOT.
      CALL TCLEAR(&1)
      CALL PLOT(20.0,10.0,-3)
      F=AMIN1(30.0/XLEN,20.0/YLEN)
      CALL FACTOR(F)
      GO TO 10
    1 CALL PLOT(3.0,3.0,-3)
      X0=0.0
      Y0=0.0
      XL=XLEN
      YL=YLEN
      GO TO 10
      ENTRY ORG(XLEN,YLEN)
C---------ENTER HERE FOR ALL BUT 1'ST CALL TO THIS ROUTINE.
      CALL TCLEAR(&2)
      GO TO 10
    2 VX=5.0+XL
      VY=5.0+YL
      Y=Y0+VY+YLEN
      YL=YLEN
      IF(Y.GT.25.0) GO TO 3
      CALL PLOT(0.0,VY,-3)
      Y0=Y0+VY
      IF(XLEN.GT.XL) XL=XLEN
      GO TO 10
    3 VY=-Y0
      CALL PLOT(VX,VY,-3)
      XL=XLEN
      X0=X0+VX
      Y0=0.0
   10 RETURN
      END
      SUBROUTINE BFMR(L1,L2,N,TE,TI,NP,ALPHA,ME,MI)
C*********************************************************************
C    *   CALCULATE BEAT FREQUENCY MIXING RATES.                      *
C    *   BOTH ELECTRON & ION MOTION IS ACCOUNTED FOR.                *
```

```
C    *    ELECTRON - ION COLLISIONS ARE FULLY ACCOUNTED FOR.              *
C    *    PONDEROMOTIVE FORCE ON BOTH ELECTRONS AND IONS.                 *
C    *    RICHARD D. MILROY          AUGUST, 14, 1977.                    *
C*************************************************************************
      COMPLEX J/(0.0,1.0)/,KE(1000),KI(1000),EP(1000),EDE(1000)
     1,EDI(1000),ECE(1000),ECI(1000),JE(1000),JI(1000),X,B(1000)
      REAL L1(1000),L2(1000),N(1000),TE(1000),TI(1000),W3(1000),K3(1000)
     1,W1(1000),W2(1000),ALPHA(1000),K1,K2,NE(1000),NI(1000)
      C=3.0E+10
C------- FIND W3 AND K3.
      DO 1 I=1,NP
      KK=I
      W1(I)=1.885E+15/(L1(I))
      W2(I)=1.885E+15/(L2(I))
      W3(I)=W1(I)-W2(I)
      WPE=5.63E+4*SQRT(N(I))
      IF(WPE.GE.W2(I)) GO TO 8
      K1=SQRT(W1(I)**2-WPE**2)/C
      K2=SQRT(W2(I)**2-WPE**2)/C
    1 K3(I)=K1+K2
      GO TO 9
    8 NP=KK-1
C------- FIND LINEAR SUSCEPTABILITIES.
    9 CALL LSUS(W3,K3,N,TE,TI,NP,KE,KI)
C-------- CALCULATE COLLISIONAL CORRECTION FACTOR.
      DO 7 I=1,NP
      VEI=2.53E-05*N(I)/(TE(I)**1.5)
      WPE=5.63E+4*SQRT(N(I))
      B(I)=1.0-J*W3(I)*VEI/(WPE**2)
      KE(I)=B(I)*KE(I)
      KI(I)=B(I)*KI(I)
    7 CONTINUE
C------- SPECIFY DIELECTRIC CONSTANT EP.
      DO 2 I=1,NP
    2 EP(I)=1.0+KE(I)+KI(I)
C------- SPECIFY PONDEROMOTIVE 'FIELDS'.   FOR I1=I2=1.0
      DO 3 I=1,NP
      WPE2 = 3.17E+09*N(I)
      C1 = C*SQRT(1.0-WPE2/(W1(I)**2))
      C2 = C*SQRT(1.0-WPE2/(W2(I)**2))
      X=K3(I)*L1(I)*L2(I)*J*C/SQRT(C1*C2)
      EDE(I)=3.11E-16*X
    3 EDI(I)=-1.69E-19*X
C------- SPECIFY COULOMB FIELDS.
      DO 4 I=1,NP
      ECE(I)=-KE(I)*(EDE(I)*(1.0+KI(I))-KI(I)*EDI(I))/(EP(I)*B(I))
    4 ECI(I)=-KI(I)*(EDI(I)*(1.0+KE(I))-KE(I)*EDE(I))/(EP(I)*B(I))
C------- SPECIFY ELECTRON AND ION CURRENTS.
      DO 5 I=1,NP
      X=J*W3(I)/12.566
      JE(I)=X*ECE(I)
    5 JI(I)=X*ECI(I)
C------- SPECIFY BEAT FREQUENCY MIXING RATES.
      DO 6 I=1,NP
      X=2.0E-7*W1(I)/W3(I)
      A31E = X*REAL(EDE(I)*CONJG(JE(I)))
      A32E = X*REAL(ECI(I)*CONJG(JE(I)))
      A31I = X*REAL(EDI(I)*CONJG(JI(I)))
      A32I = X*REAL(ECE(I)*CONJG(JI(I)))
      A3E = A31E+A32E
```

```
      A3I = A31I+A32I
      ALPHA(I) = A3E+A3I
      NE(I) = K3(I)*2.0*CABS(BCE(I))/(6.03E-09*N(I))
      NI(I) = K3(I)*2.0*CABS(BCI(I))/(6.03E-09*N(I))
C     WRITE(6,601) L2(I),A31E,A32E,A3E,A31I,A32I,A3I,ALPHA(I)
  601 FORMAT(' ',1P8E12.4)
    6 CONTINUE
      RETURN
      END
      SUBROUTINE LSUS(W3,K3,N,TE,TI,NP,KE,KI)
C*****************************************************************************
C     * CALCULATE LINEAR SUSCEPTABILITIES FOR BOTH ELECTRONS & PROTONS. *
C     *          RICHARD D. MILROY         AUGUST, 14, 1977.             *
C*****************************************************************************
      COMPLEX KE(1000),KI(1000),J/(0.0,1.0)/,FO
      REAL W3(1000),K3(1000),N(1000),TE(1000),TI(1000)
C------- FIND KE.
      DO 1 I=1,NP
      WP=5.63E+4*SQRT(N(I))
      VO=5.93E+7*SQRT(TE(I))
      AO=W3(I)/(K3(I)*VO)
      IF(AO.GT.50.0) GO TO 3
      KE(I)=2.0*(WP*AO/W3(I))**2*(1.0+J*AO*FO(AO))
      GO TO 1
    3 KE(I)=-(WP/W3(I))**2
    1 CONTINUE
C------- FIND KI.
      DO 2 I=1,NP
      WP=1.31E+3*SQRT(N(I))
      VO=1.38E+6*SQRT(TI(I))
      AO=W3(I)/(K3(I)*VO)
      IF(AO.GT.50.0) GO TO 4
      KI(I)=2.0*(WP*AO/W3(I))**2*(1.0+J*AO*FO(AO))
      GO TO 2
    4 KI(I)=-(WP/W3(I))**2
    2 CONTINUE
      RETURN
      END
      COMPLEX FUNCTION FO(AO)
C------- CALCULATE FO AS DEFINED ON PAGE 178 OF STIX.
      COMPLEX J/(0.0,1.0)/
      REAL FOSPL(12,5)
      DATA FOSPL/0.2,0.5,0.8,1.0,1.2,1.4,1.7,2.0,2.5,3.0,3.5,4.0,
     1 .3895,0.7199,1.064,1.076,1.015,0.913,0.7451,0.6027,0.4462,0.3565
     2,0.2992,0.2587,1.003,1.299,0.5474,-0.2165,-0.4162,-0.5635,-0.532,
     3-0.4114,-0.2316,-0.1393,-0.0931,0.0,0.0,0.987,-3.4911,-0.3284,
     4-0.6701,-0.06606,0.1709,0.231,0.1287,0.5593E-1,0.03642,0.0,1.097
     5,-4.976 ,5.272,-0.5696,1.007,0.2633,0.06673,-0.06821,-0.0485,
     6-0.01301,-0.02428,0.0/
      AOE=AO
      IF(AOE.GT.10.)AOE=10.0
      AOES=AOE*AOE
C     SPLINE APPROXIMATION FOR FO
      IF(AO.LT.0.21) GO TO 90
      IF(AO.GT.3.25)GO TO 12
      DO7 I=1,12
      IF(AO.LT.FOSPL(I,1))GO TO 8
    7 CONTINUE
    8 I=I-1
      DA=AO-FOSPL(I,1)
```

```
      PO=POSPL(I,2)+POSPL(I,3)*DA+POSPL(I,4)*DA*DA+POSPL(I,5)*DA**3
      PO=J*PO
      GO TO 11
   12 PO=J/A0*(1.0+0.5/(A0*A0)+0.75/(A0**4)+1.875/(A0**6)+6.563/(A0**8))
      GO TO 11
   90 PO=2.0*J*(A0-0.666667*A0**3+.266667*A0**5-.0761905*A0**7)
      GO TO 11
   11 DAMP=1.772*EXP(-A0ES)
      PO=PO+DAMP
      RETURN
      END
```

VITAE

NAME: Richard Douglas Milroy

DATE AND PLACE OF BIRTH: March 20, 1951, Hardisty, Alberta, Canada

DEGREES:

B.Sc. Physics (Honors)	Univ. of Alberta	April, 1973
M.Sc. Elect. Eng.	Univ. of Alberta	March, 1976
Ph.D. Elect. Eng.	Univ. of Alberta	Sept., 1978

WORK EXPERIENCE:

Geophysicist;	Geophysical Services Inc.	640-12 Avenue, S.W. Calgary, Alberta May 1973 to Sept., 1974
Part-time Teaching Assistant	Univ. of Alberta	1974 - 1978

AWARDS:

University of Alberta Honor Prize	1969
National Research Council of Canada Postgraduate Scholarship	1974, 1975, 1976, 1977
National Research Council of Canada Postgraduate Fellowship	1978

PUBLICATIONS:

R.D. Milroy, C.E. Capjack, C.R. James and J.N. McMullin, "MHD Simulation of a Beat Frequency Heated Plasma." Can. J. Physics, 54, No. 21, 2140-2146, 1976.

R.D. Milroy, C.E. Capjack, and C.R. James, "A Plasma-Laser Amplifier in the 11-16 μm Wavelength Range." Plasma Physics, 19, No. 10, 989-995, 1977.

C.R. James, R.D. Milroy, and C.E. Capjack, "An Infrared Plasma Parametric Amplifier." Physics & Chemisrry of Material Processing, N AƏT-BO "HaYKa" (In Press. Accepted September 1977).

J.N. McMullin, R.D. Milroy, and C.E. Capjack, "A Two-Dimensional Magnetic Flux Shell Model for Magnetohydrodynamic Simulations." (Submitted to Phys. Fluids)

R.D. Milroy, J.N. McMullin, and C.E. Capjack, "Two-Dimensional Hydrodynamic Simulations of a Laser Heated Gas Target Plasma." (In Preparation - To be submitted to Can. J. Phys.)

R.D. Milroy, C.E. Capjack, and C.R. James, "Plasma Parametric Amplification of Ultra-Short Laser Pulses." (In Preparation - To be submitted to Can. J. Phys.)

PATENT

A United States patent is being applied for through Canadian Patents and Development Limited for "Parametric Amplifier for Amplifying Laser Radiation", R.D. Milroy, C.E. Capjack, and C.R. James.