# REVERSE CURRENT IN SOLAR FLARES

by

Joshua W. Knight III



National Aeronautics and Space Administration
Grant NGL 05-020-272

Office of Naval Research
Contract N00014-75-C-0673

SUIPR Report No. 752

August 1978



INSTITUTE FOR PLASMA RESEARCH
STANFORD UNIVERSITY, STANFORD, CALIFORNIA

# REVERSE CURRENT IN SOLAR FLARES

by

Joshua W. Knight, III*

National Aeronautics and Space Administration

Grant NGL 05-020-272

Office of Naval Research

Contract N00014-75-C-0673

SUIPR Report No. 752

August 1978

Institute for Plasma Research
Stanford University
Stanford, California

*Also Department of Applied Physics

ABSTRACT

An idealized steady state model of a stream of energetic electrons neutralized by a reverse current in the pre-flare solar plasma is developed. These calculations indicate that, in some cases, a significant fraction of the beam energy may be dissipated by the reverse current. Joule heating by the reverse current is a more effective mechanism for heating the plasma than collisional losses from the energetic electrons because the Ohmic losses are caused by thermal electrons in the reverse current which have much shorter mean free paths than the energetic electrons.

Analysis of the steady state model indicates that it can not adequately describe the interaction of the beam with the solar plasma because the atmosphere is rapidly heated. If the time scale for this heating is short enough, the density of the atmosphere can be taken constant in time. The charge separation required to drive the reverse current is expected to respond to changes on a time scale very short compared to the time for the ambient plasma temperature to change significantly, so it is a reasonable approximation to use the steady state results for the electric field. With these simplifications, the heating due to reverse currents is calculated for two injected energetic electron fluxes. For the smaller injected flux, the temperature of the coronal plasma is raised by about a factor of two. The larger flux causes the reverse current drift velocity to exceed the critical velocity for the onset of ion-cyclotron turbulence, producing anomalous resistivity and an order of magnitude increase in the temperature. The heating is so

rapid that the lack of ionization equilibrium may produce a soft x-ray and EUV pulse from the corona.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Contd.)

## ACKNOWLEDGMENT

The work presented in this dissertation was initiated and directed by Professor Peter A. Sturrock. Much of the credit for any merit this work may have must therefore go to him. In addition, I wish to thank him for his help and guidance in the other research efforts I have undertaken during my tenure as a graduate student. I am indebted to Professor Sturrock as well as Professors Vahe Petrosian and Arthur B. C. Walker, Jr. for critically reading this dissertation. I would like to thank Professor Oscar Buneman for many helpful discussions about computational physics.

I have benefited from many helpful discussions with Dr. William M. Adams, Dr. Spiro K. Antiochos, Kile Baker, Dr. Roger A. Dana and Dr. Charles E. Newman. During the course of this work I have had many stimulating discussions with Dr. David D. Barbosa, Dr. Robert J. Barker, Dr. Christopher W. Barnes, Steve Langer, Keith A. Marzullo, Dr. Ronald N. Moore, Dr. David Roberts, Hal Tompkins, Rick Trebino and Steve Turk.

I would like to thank Jane Johnston for carefully typing this thesis. My wife, Mary Ann Knight, has contributed immeasurably to the completion of this work through her support both moral and financial.

Finally, I gratefully acknowledge the financial support provided by the National Aeronautics and Space Administration under Grant NGL 05-020-272 and the Office of Naval Research under Contract N00014-75-C-0673.

# 1. INTRODUCTION

This dissertation examines the consequences of reverse currents that may be expected to develop in the solar atmosphere in response to the imposition of a directed stream of energetic (non-thermal) electrons. The phenomena which indicate the presence of streams of electrons manifest themselves primarily in the "flash phase" of solar flares (Svestka 1975). Not all flares exhibit a "flash phase" (Svestka 1975, Sweet 1969, Sturrock and Coppi 1966) and the existence of directed streams of non-thermal electrons is not universally accepted (Svestka 1975, Brown 1974, Brown and Melrose 1977). A short historical review is presented (cf. 1.1) as an attempt to place the phenomena in perspective. Observations that indicate the presence of energetic electrons in the solar atmosphere are reviewed and the introduction concludes with a short summary of our present theoretical understanding of the flare process.

In Chapter 2 the objections to unneutralized electron beams and previous work on reverse currents are summarized and a steady state model of a stream of energetic electrons neutralized by a reverse current is developed. In Chapter 3 the model is modified to include time dependence for a restricted case. The results of Chapters 2 and 3 are summarized in Chapter 4 and possible extensions of the present work are suggested. Details of the numerical calculations of Chapters 2 and 3 are discussed in the appendices.

## 1.1 Historical Overview

The sun is the closest star to the earth and the only star which we can presently observe in great detail. Aside from the intrinsic interest of solar phenomena, we can hope that by understanding solar phenomena we will gain insight into what is likely to happen on other stars like the sun. The sun is a normal G type main sequence star, but by virtue of its position it is the brightest object in the sky. The importance of the sun to life on earth cannot be overstated. In the introduction to his book, The Sun, C. A. Young (1902a) emphasizes this point.

> "It is true from the highest point of view the sun is only one of a multitude - a single star among millions - thousands of which, most likely, exceed him in brightness, magnitude and power. He is only a private in the host of heaven.
>
> "But he alone, among the countless myriads, is near enough to affect terrestrial affairs in any sensible degree; and his influence upon them is such that it is hard to find the word to name it; it is more than mere control and dominance. He does not, like the moon, simply modify and determine certain more or less important activities upon the surface of the earth, but he is almost absolutely, in a material sense, the prime mover of the whole. To him we can trace directly nearly all the energy involved in all phenomena, mechanical, chemical or vital. Cut off his rays for even a single month, and the earth would die; all life upon its surface would cease."

The great preponderance of the energy flux from the sun is, to the best of our knowledge, very nearly constant (Smith and Gottlieb 1974). It is only in those portions of the electromagnetic spectrum where the solar output is small (radio XUV, X-ray), in individual spectral lines (e.g. $H_\alpha$, Ca H and K), and in particle emission (the solar wind, energetic electrons and nuclei), that the sun's output varies significantly due to solar activity.

The most obvious manifestations of solar activity are sunspots. Sunspots have been observed telescopically since 1611, shortly after the invention of the telescope, and with the unaided eye on infrequent occasions since ancient times (Bray and Loughhead 1964). It is not clear which of four men, Galileo Galilei, Johann Goldsmid, Thomas Harriot or Christopher Scheiner, actually made the first telescopic observation of sunspots (Bray and Loughhead 1964). That another manifestation of solar activity, faculae, were observed at about the same time is demonstrated by the title of Christopher Scheiner's (1630) book, Rosa Ursina Sive Sol ex Admirando Facularum and Macularum Fuarum Pheonomeno Varius (see Eddy et al. 1977, Meadows 1970). In the first half of the 19th century Schwabe (1844) announced the possible existence of the sunspot cycle with a period of about 10 years ("von ungefahr 10 Jahren"). Wolf (1852) later deduced a more accurate period of 11.1111 ± .038 years or "de sorte que neuf periodes equivalent justement a un siecle". Wolf (1852) also deduced from earlier records the years of sunspot minima back to 1700, but the earlier portion of this historical reconstruction has been questioned recently (Eddy 1976).

The first recorded observation of a solar flare occurred on September 1st, 1859. A relatively rare "white light flare", visible against the photosphere, was simultaneously observed by Carrington (1859) and Hodgson (1859). In 1868 Janssen (1869) and Lockyer (1869) independently discovered that prominences could be seen outside eclipse with a spectroscope with a wide entrance slit. Thereafter various observers, especially Secchi (1877) made extensive visual observations of the forms of the chromosphere and prominences using this technique. Flares in individual lines were observed quite often from the 1870's onward (see Young 1871, 1902b,c for early examples). The first photographs of flares were obtained by Hale (1892) with a spectroheliograph of his own invention (Hale 1891). Deslandres (1893) independently developed a similar instrument, and the basic principle of the spectroheliograph was known to Janssen (1869) who actually constructed an instrument similar to the spectrohelioscope (Millochau and Stefanik, 1906) for observing prominences but abandoned it in favor of a widened spectroscope slit. The basic principle was independently discovered by Braun, and Lohse attempted the construction of a spectroheliograph (Hale 1906). The matter of who actually used a "spectroheliograph" first was the subject of some debate between Deslandres and Hale (Hale 1906, Deslandres, 1905) but this distinction is generally given to Hale. In 1908 Hale (1908) made the first observation of magnetic fields on the sun, and realized soon thereafter that magnetic fields, sunspots and flares were intimately connected (Hale 1929). Because the spectroheliograph took a relatively long time to form an image of the whole sun, the systematic investigation of flares did not begin until the spectrohelioscope, constructed by Hale in 1926 (Hale 1929), was fully developed (Smith and Smith 1963). The development of the polarizing monochromatic filter (Lyot filter) by Ohman in 1938 (Ohman 1938), independently of Lyot's original proposal (Loyt 1933, Evans 1949), allowed photographs of the entire solar disk in one spectral line to be made rapidly. This type of filter is still widely used in flare patrol telescopes and solar observatories.

Jansky (1933) made the first observation of radio emission from an extra-terrestrial source. It was not until 1942 that Hey (1946) discovered meter wavelength radiation from the sun. At about the same time Southworth (1945) discovered centimeter wavelength radiation from the sun. Reber (1944) made the first published report of radio emission from the sun; the earlier work was not published due to its association with the war effort. Appleton (1945) published evidence for radio emission from the sun in the 7-30 meter wavelength band. Appleton's results were based on amateur radio operators' reports (dating from 1936) of "hiss" heard only during the daylight hours and frequently before sudden fade outs. Appleton and Hey (1946) noted that some radio bursts were associated with flares. Covington (1948) first reported microwave bursts from the sun near the maximum of solar cycle 18.

Burnight (1949) reported the first observation of X-ray emission from the sun. Burnight's observation was made using photographic film with aluminum and beryllium filters flown in a captured V2 rocket. Peterson and Winkler (1958) made the first observation of a flare associated impulsive X-ray burst using a balloon borne proportional counter.

## 1.2 Review of Observations

The presence of energetic electrons in the solar atmosphere is inferred from impulsive hard X-ray bursts, impulsive microwave bursts and observations of energetic electrons by satellites in earth orbit. Impulsive microwave bursts are rapid enhancements of radio flux at frequencies greater than ~ 1 GHz. These impulsive enhancements occur simultaneously with impulsive X-ray and EUV bursts and often show very similar time structure, even down to the fine details of the time profiles (Peterson and Winkler 1959, Kundu 1961, Anderson and Winkler 1962, Kane and Donnelly 1971, deFeiter 1975, Svestka 1975). The impulsive microwave bursts are generally attributed to gyro-synchrotron radiation from electrons with energies greater than ~ 100 keV (Holt and Ramaty 1969, Svestka 1975). The gradual post-burst increases can be interpreted as thermal bremsstrahlung from the flare-associated soft X-ray plasma and are usually accompanied by radio emission at lower frequencies (Svestka 1975). The apparent discrepancy between the number of electrons required to produce the impulsive X-ray bursts and the number of electrons required to produce the impulsive microwave emission (Peterson and Winkler 1959) can be resolved if the details of the microwave emission in the solar atmosphere are considered (Holt and Ramaty 1969, Takakura 1972). The generation and propagation of microwaves in the solar atmosphere during a solar flare are complicated processes involving the magnetic field configuration, ambient plasma density and temperature and density and energy spectrum of the non-thermal electrons (Holt and Ramaty 1969, Kruger 1972, Takakura 1972, Svestka 1975). Therefore, it is difficult to unambiguously infer the number and energy spectrum of the non-thermal electrons from the observed microwave emission.

In some flares, non-thermal electrons escape into the interplanetary medium and are observed by satellites in earth orbit (Svestka 1975, Lin 1974). Since the electrons apparently propagate primarily along magnetic field lines in the interplanetary medium, electrons are observed primarily from flares in the western half of the visible hemisphere of the sun or from flares behind the west limb of the sun (Svestka 1975, Lin 1974). Lin (1974) concludes that there are two distinct types of non-relativistic electron bursts ($E < 500$ keV) observed at 1 AU, "pure electron events", that is those not accompanied by energetic ($> 10$ mev) protons, and "mixed events" during which both energetic electrons and protons are observed. The energy spectra of the "pure electron" events can be well fitted between 5 keV and 100 keV by a power-law in energy, $dN/dE \propto E^{-\gamma}$, with $\gamma \sim 2.5$-$5.5$ but exhibit a rapid steepening at energies above 100 keV (Lin 1974). On the other hand the typical spectra of energetic electrons for "mixed events" extend smoothly in a power-law out to and beyond 10 mev and tend to be somewhat harder ($\gamma \sim 2.5$-$4.5$) (Lin 1974). When impulsive X-ray bursts are associated with electrons observed at 1 AU, $10^2$-$10^3$ more electrons are required to produce the impulsive hard X-ray bursts than escape to the interplanetary medium (Lin and Hudson 1971, Lin 1974).

It is now generally believed that the mechanism for the production of impulsive hard X-ray bursts is bremsstrahlung from electrons scattering on protons and heavier ions (Kane 1974, Svestka 1975, Brown 1975). Smaller impulsive events generally consist of one or a few spikes with comparable e-folding rise and fall times of $\sim$ 10 s (Kane 1969, Kane and Anderson 1970, Crannell et al. 1977). Larger events, with total durations of minutes or tens of minutes, usually have a complex spiky time structure (Svestka 1975, Hoyng et al. 1976). Frost and Dennis (1971) and Frost (1974) have also reported an apparently distinct non-impulsive non-thermal hard X-ray component in some larger events, after the impulsive phase of the flare and possibly associated with a second phase of particle acceleration. In this work, we restrict our attention to the impulsive hard X-ray bursts, and assume that both the later "second phase" hard X-rays and the "gradual components" in the low energy channels ($\lesssim$ 50 keV) of some instruments are distinct phenomena.

The spectral information on impulsive hard X-ray bursts is limited, but most events can be reasonably fitted to a decreasing power-law in photon energy between 10-20 keV and 100-150 keV (Kane 1974, Brown 1975, Hoyng 1977). The power-law index is typically between 2.5 and 5 (Kane 1974, Svestka 1975) although some bursts have very soft spectra and power-law indices as large as 8 have been reported (Peterson et al. 1974).

Most events show a softening of the spectrum at higher energies (Kane 1974, Svestka 1975). This bend or "knee" in the power-law spectrum usually occurs between 60 and 100 keV (Brown 1975, Svestka 1975) but in some events can occur as high as 500 keV (Brown 1975). Since the high energy cut-off of many instruments is below 500 keV [e.g. OSO-7 (Peterson et al. 1974), OSO-5 (Frost et al. 1970) or OGO-5 (Kane and Anderson 1970)] such a break in the spectrum may be present in many events for which no break is reported. It is obvious that the power-law must flatten at low energies, otherwise the total X-ray flux would diverge. However, the determination of the low energy cut-off is difficult because the X-ray emission at low energies (< 10 keV) is dominated by the gradual quasi-thermal component in most events (Brown 1975, Svestka 1975).

Although the interpretation of the X-ray spectrum as bremsstrahlung from a non-thermal (i.e. non-Maxwellian in energy) distribution of electrons is widely accepted, some workers advocate a thermal interpretation for many impulsive X-ray bursts (for example Chubb 1970, Elcan 1976, Crannell et al. 1977) and some events seem to fit an exponential rather than a single power-law spectrum (Elcan 1976, Crannell et al. 1977). However, the spectral data are poor, particularly at higher energies (primarily due to counting statistics), and it is not clear that an exponential spectrum is to be preferred over two power-laws or some other form for the spectra, Brown (1974) has demonstrated that any observed hard X-ray spectrum can be produced by a thermal plasma with a suitable temperature distribution in the source. Brown (1975) has also pointed out that the emitted X-ray spectrum is rather insensitive to the source electron energy spectrum and concludes that a power-law electron spectrum is not strongly mandated by the presently available data.

There are theoretical objections to multi-thermal models of impulsive hard X-ray bursts (Kahler 1975) and models that produce power law spectra by superposing different exponential spectra seem somewhat contrived to this author despite assertions to the contrary by some workers (e.g. Brown 1975). In more recent "thermal" models of impulsive hard X-ray bursts (e.g. Smith and Lilliequist 1978), the electron distribution is not expected to be Maxwellian. Neither the theoretical objections (Kahler, 1975) or the limited observational support for thermal electron distributions (Elcan 1976, Crannell et al. 1977) are relevant to this type of model.

There is some support, from observations of impulsive EUV bursts, for the view that the impulsive hard X-ray bursts are produced by non-thermal, energetic electrons streaming from the corona to the chromosphere. Impulsive EUV bursts have been observed directly by satellites (for example Kelley and Rense 1972, Hall 1971) and indirectly from the ionospheric effects they produce (Donnelly 1974). These bursts show close time coincidence with the impulsive X-ray and microwave bursts and the time profiles closely resemble the X-ray and microwave bursts (Noyes 1974, Donnelly 1974, Kane and Donnelly 1971, Kane 1974). The energy radiated in the 10-1030 Å band is $\gtrsim 10^4$-$10^5$ times the energy radiated in the associated impulsive hard X-ray burst (Donnelly 1974, Kane and Donnelly 1971). This ratio of energy radiated in the EUV and hard X-ray bands corresponds qualitatively with the expected ratio of

Coulomb collisional losses to bremsstrahlung emission from a thick-target hard X-ray source (Koch and Motz 1959, Petrosian 1973, Donnelly 1974, Brown 1975). There are indications that the EUV radiation originates low in the chromosphere. The density of the EUV emitting region can be estimated to be $\geq 10^{12}$ $cm^{-3}$ (Donnelly 1974, Kane and Donnelly 1971), corresponding to the solar chromosphere, and the EUV bursts exhibit (statistical) limb darkening which would be expected if the radiation originates in the chromosphere (Kane and Donnelly 1971). Although the observations presently available do not exclude other interpretations, the preponderance of evidence seems to favor bremsstrahlung from a non-thermal distribution of energetic electrons as the source of impulsive hard X-ray bursts (Svestka 1975).

If the emergent X-ray spectrum were known exactly, the spectrum of the energetic electrons that produce the radiation, averaged over the source, could in principle, be recovered (Brown 1975). The two extreme approximations that are usually considered are "thick-target" and "thin-target" (Brown 1975, Svestka 1975, Hudson 1974). In the thin-target approximation the electrons lose a negligible amount of energy in the hard X-ray source (Brown 1975, Svestka 1975, Hudson 1974). In this approximation, the mean electron source spectrum [i.e. the instantaneous average of the electron energy spectrum over the emitting volume weighted by the background density, see Brown (1975)] is just the spectrum of accelerated electrons. In the thick-target approximation, this is not the case.

In the thick-target approximation, the electrons lose all their energy (primarily by Coulomb collisions) in the source region. Since the mean free path of the more energetic electrons is longer, the energy spectrum of the electrons averaged over the emitting volume is harder. If we assume the density is uniform in the source, that the electrons are all streaming downward and that the accelerated electron energy spectrum is a fairly steep power-law, then the approximate difference between the inferred mean electron source spectrum in the thin and thick-target approximations can be estimated. In this case, since the mean free path of an electron against Coulomb collisions is approximately proportional to $E^2$, the effective source volume for electrons of energy $E$ in the accelerated spectrum is also approximately proportional to $E^2$. Since the injected spectrum is very steep (by assumption), once an electron has lost an appreciable fraction of its energy, it no longer contributes significantly to the emergent X-ray flux. Therefore, to produce the same power-law index of emergent X-rays the index of the injected electron beam must be $\sim 2$ greater (a softer injected spectrum) in the thick-target case than in the thin-target approximation. The preceeding simple analysis neglects beaming effects in the case the energetic electron velocity distribution is anisotropic (Petrosian 1973, Brown 1972) and the exact behavior of the Coulomb cross section. However, the conclusion is found to be qualitatively correct in thick-target models of impulsive hard X-ray bursts for X-ray energies below $\sim 100$ keV even when a more detailed analysis is performed (Brown 1975, Petrosian 1973, Hudson 1972, Brown 1971). The more detailed calculations indicate that, depending on the assumptions and model characteristics, thick-target models require injected electron power-law indices $\sim 1.5$-2 greater than thin-target models for the same emergent X-ray spectra.

Some early models of impulsive X-ray bursts considered impulsive injection of the energetic electrons and the subsequent decay of the impulsively injected electrons in the source region (e.g. Takakura and Kai 1966). In its simplest form this model does not agree with observations since it predicts a systematic hardening of the burst spectra during the decay of the hard X-ray burst (Brown 1975, Petrosian 1973) for the same reason that the source averaged energetic electron spectrum is harder in the thick-target models, i.e. the low energy electrons lose their energy more rapidly than the high energy electrons. Brown (1972) has introduced a modification of the usual coronal impulsive hard X-ray source that removes this particular objection, but it requires the assumption of an average source density that is energy dependent ($\bar{n} \propto E^{\alpha}$, $\alpha > 3/2$). Brown (1972) motivates this assumption by invoking an energy dependent pitch angle distribution for the accelerated electrons, but the simplicity of the original "coronal trap" model is lost. This type of model can explain the observation of impulsive X-ray bursts from "behind-the-limb" flares since portions of the X-ray source are high in the corona. However, since behind-the-limb flares also produce high energy X-rays, this model requires densities $\geq 10^{11}$ cm$^{-3}$ high ($\geq 10^{9}$ cm) in the solar atmosphere. If this were the case, impulsive X-ray bursts from behind-the-limb flares could be explained by thick-target models as well. Since the product of the instantaneous number of energetic electrons in the source and background density determines the emergent X-ray flux, Brown's (1972) model requires a much larger number of energetic (> 20 keV) electrons than equivalent thick-target models. Additionally, since most of the energy resides in the low energy electrons which encounter only low densities ($\approx 10^9$), these electrons cannot be invoked to account for the impulsive EUV bursts which are emitted from regions where the density is $\geq 10^{12}$ $cm^{-3}$ (Donnelly 1974) and which are observed simultaneously with impulsive X-ray bursts (this is also true of more recent "thermal" models, e.g. Smith and Lilliequist 1978).

Aside from some difficulty in accounting for impulsive X-ray bursts from behind-the-limb flares, thick-target models for the hard X-ray bursts are at least not excluded by present observations. Since they have the advantage of also providing the energy required for the impulsive EUV bursts (Donnelly 1974), it seems reasonable to accept the thick target approximation for the production of the hard X-ray bursts. In this case, since the time for the electrons to lose all their energy is short compared to the time scale of the impulsive X-ray burst (Brown 1975, Petrosian 1973), variations in the X-ray flux and spectrum are attributed to changes in the (unspecified, c.f. 1.3) acceleration process.

In the thick-target model the energy flux of the electron stream required to produce a specified X-ray flux at 1 AU depends on (a) the anisotropy of the electron velocity distribution, (b) the power-law index of the X-ray flux and (c) the lowest energy to which the power-law in energy is assumed to extend for the energetic electrons (Brown 1975, Petrosian 1973). Neglecting possible beaming of the bremsstrahlung radiation (Petrosian 1973, Brown 1972) and backscatter from atmosphere (Langer and Petrosian 1977), we can obtain an order of magnitude estimate for the flux of non-thermal electrons at the sun for an observed flux of X-rays at 1 AU. If the flux of X-rays at some energy $E_0$ at earth is $\mathcal{F}$ (photons $cm^{-2}$ $s^{-1}$ $keV^{-1}$), then the total X-ray photon flux is $10^{27.45}\ \mathcal{F}$ photons keV$^{-1}$ s$^{-1}$ . Since the observed power-law spectra are typically fairly steep (Brown 1975, Svestka 1975), most of the X-rays at $E_0$ are produced by electrons with only slightly higher energies. The total efficiency (the ratio of bremsstrahlung losses to Coulomb collisional losses) is approximately $10^{-6}E$ for a thick-target hydrogen plasma (Koch and Motz 1959, Petrosian 1973). Therefore, the total non-thermal electron flux in the source above $E_0$ must be $\approx 10^{33.45}\,\mathcal{F}\,E_0^{-1}$ keV$^{-1}$ .

## 1.3 Review of Flare Theories

In the preceeding sections we have discussed some of the observed properties of solar flares as they relate to the inferred presence of non-thermal electrons in the solar atmosphere during a flare. We have not dealt with most of the diverse phenomena associated with solar flares. Svestka (1975) lists thirty-seven "basic properties of flares". When all the subtopics are counted, Svestka's list contains more than eighty observational aspects of flares. With such a large number of properties to be considered, it is not surprising that a wide variety of flare theories and models have been proposed. Since reviews of flare theories exist in the literature (Svestka 1975, Sweet 1969), the selection of theoretical ideas discussed here is only representative and not exhaustive. This discussion of flare theories is included only to show how the production of non-thermal electron streams fits in the present theoretical picture of solar flares and therefore no particular model will be treated in detail.

It is now widely believed that the energy released in solar flares is stored in the magnetic fields in the upper solar atmosphere (Rust 1977, Svestka 1975, Sweet 1969). The energy which is available for release is the excess energy of the non-potential magnetic field configuration above the energy in the potential (current-free) field (Rust 1977, Svestka 1975, Sweet 1969). Because the magnetic field energy density is generally believed to be greater than the thermal energy density of the plasma in the upper solar atmosphere, the non-potential field configurations must be nearly force-free (Gold and Hoyle 1960, Sturrock 1974). Although many non-potential field configurations have been proposed, these configurations can be divided into two broad categories depending on the distribution of currents in the solar atmosphere (Svestka 1975, Sturrock 1974). One possibility is a force-free configuration in the form of twisted flux tubes (Gold and Hoyle 1960, Alfven and Carlqvist 1967, Spicer 1977) or sheared field lines (Tanaka and Nakagawa 1973). In this case the currents are distributed over a large volume in the atmosphere. The other possibility is that the field is largely current-free with the current concentrated in current sheets (Sweet 1958, Syrovatsky 1966, Sturrock 1968, Priest and Heyvaerts 1974). A large number of flare models have been developed under the assumption that current sheets develop in the solar atmosphere as a result of motions in the photosphere or the emergence of new flux (Svestka 1975, Sweet 1969). Barnes and Sturrock (1972) have studied the development of non-potential force-free fields due to photospheric motions and found that the stored energy in the force-free configuration can exceed that of a configuration with a current sheet. They concluded that one possible sequence of events that would produce a current sheet in the solar atmosphere was the conversion of a more energetic force-free configuration to a configuration with a sheet. Priest and Heyvaerts (1974) examined the production of a current sheet when new flux emerges into a pre-existing magnetic field configuration.

The earliest electromagnetic models of flares invoked the production of non-thermal electrons and realized the importance of electric fields at "neutral points" in the magnetic field (Giovanelli 1946, 1947, 1948, Hoyle 1948). Dungey (1958) pointed out that, when reconnection of magnetic field lines occurs, a DC electric field will be developed in the reconnection region which could lead to acceleration of charged particles. In "current interruption" models (Alfven and Carlqvist 1967) electrons are accelerated by the DC electric field that develops when the "inductive circuit" is opened. In models in which reconnection occurs in a current sheet (Sturrock 1968, Friedman and Hamberger 1969, Coppi and Friedland 1971), some acceleration by a DC electric field at the neutral point may occur, but the bulk of the acceleration is usually attributed to stochastic acceleration of electrons by high frequency electric fields that develop during the reconnection process due to plasma instabilities (Sturrock 1974, Smith 1974). It has proved difficult to develop a self consistent theoretical model of the rapid acceleration of the number of electrons required to produce the observed X-ray flux (Smith 1977a, b, Brown and Melrose 1977). At present, the mechanism by which electrons are accelerated in the impulsive phase of solar flares is not well understood theoretically (Svestka 1975). However, simple considerations indicate that if the energy stored in the magnetic field is released in the low density corona, particles can be expected to acquire energies of 10-100 keV (Sturrock 1974). Furthermore, the ingredients of many possible acceleration mechanisms (DC electric fields, plasma turbulence) are natural by-products of most processes which release the energy stored in the magnetic field. Therefore, since there is observational evidence for the acceleration of electrons in the impulsive phase of flares, we will assume that this acceleration does occur even though the exact mechanism has yet to be elucidated.

## 2. STEADY STATE MODEL OF BEAM AND REVERSE CURRENT

### 2.1 Objections to Unneutralized Beams

The simplest thick-target model for the production of impulsive X-ray bursts is that considered by Petrosian (1973). In this model a beam of energetic electrons is assumed to propagate from the corona to the chromosphere. All the electrons are assumed to have their velocities in the same direction until they lose all their energy, approximating a source in which the energetic electrons stream down a nearly vertical magnetic field line with small pitch angles into an atmosphere with a small density scale height (Petrosian 1973). Several authors (Brown 1976, Brown and Melrose 1977, Colgate _et al._ 1977, Hoyng 1977, Hoyng _et al._ 1976) have pointed out difficulties if this electron stream is not neutralized by a reverse current.

Brown (1976) pointed out that the number of electrons required to stream from the corona to the denser portions of the solar atmosphere during some impulsive hard X-ray bursts was quite large. Indeed in some events as large as $10^{39}$ (Hoyng _et al._ 1976), or all the electrons in the solar atmosphere above the level where the electron density is $\sim 10^{13}$ cm$^{-3}$ (Brown 1976). Another objection to the existence of an unneutralized beam is that the magnetic energy that would be stored in this beam is many orders of magnitude larger than the total flare energy (Colgate _et al._ 1977). If $N$ is the total number of electrons streaming downward over the duration $\tau$(s) of the impulsive phase, the magnitude (emu) of the current may be estimated from

$$I \simeq ec^{-1} N\tau \qquad (2.1)$$

If the transverse and longitudinal dimensions of the stream are of order $R$ (cm), an estimate of the strength $B$ (gauss) of the magnetic field produced by the stream is given by

$$B \simeq 2IR^{-1} ; \qquad (2.2)$$

and the total energy $U$ (ergs) of this magnetic field may be estimated from

$$U \simeq \frac{1}{8\pi} R^3 B^2 \simeq \frac{1}{2\pi} I^2 R , \qquad (2.3)$$

which becomes

$$U \simeq \frac{1}{2\pi} e^2 c^{-2} N^2 T^{-2} R \simeq 10^{-40.4} N^2 T^{-2} R . \qquad (2.4)$$

Kane and Anderson (1970) estimate the total energy involved in a typical small flare to be $\sim 10^{29}$ ergs, the time scale to be $\sim 10^2$ s, and the characteristic length scale to be $\approx 10^{8.5}$ cm and infer from the X-ray data that the total number of energetic electrons is $\approx 10^{35}$. For these values the above formulae lead to estimates of $I \approx 10^{13.2}$, $B \approx 10^5$, and $U \approx 10^{34}$. For a large event the total flare energy could be $\approx 10^{32}$ ergs, the length scale $\approx 10^{9.5}$, the characteristic time $\approx 10^3$ and the total number of energetic electrons $\approx 10^{39}$ (Hoyng et al. 1976). In this case $I \sim 10^{16.2}$, $B \approx 10^7$ and $U \approx 10^{41}$. Clearly a model which involves an unneutralized beam leads to unacceptably high values of the magnetic field and magnetic energy associated with the beam.

Problems associated with the propagation of high current beams of charged particles not neutralized by a reverse current have been considered in other contexts. Alfven (1939) examined the limitations on the propagation of electrostatically neutralized high current beams of relativistic charged particles, motivated by an apparent sidereal day variation in the cosmic ray flux (Alfven 1938, Compton and Getting 1935), which later proved to be spurious (Dorman 1974). Consider a cylindrically symmetric, mono-energetic, uniform beam of charged particles moving through a background of opposite charge so distributed that the charge density (esu) is everywhere zero. If the beam is infinite in extent along the symmetry axis and has a radius of $R$, then the magnetic field as a function of distance from the axis for $r \leq R$ is

$$B(r) = \frac{2I(r)}{r} = 2\pi jr , \tag{2.5}$$

where $I(r)$ is the current inside $r$ and $j$ is the current density (assumed uniform). The gyro radius of a charged particle in a magnetic field is

$$r_g = \frac{pc}{qB} \tag{2.6}$$

where $p$ is the particle momentum and $q$ is the particle charge. Consider a test particle of the same charge and mass as the beam particles moving in the magnetic field of the beam. Suppose the test particle is initially at the outer edge of the beam $(r=R)$ and has the same momentum as the beam particles. We denote by $R_A$ the beam radius for which the gyroradius of this particle in the average field it sees in its trajectory is equal to the beam radius. For a beam of this radius $(R_A)$, the particle will cross the axis of symmetry with its momentum perpendicular to that of the beam particles. If the radius of the beam is increased, the particle will cross the symmetry axis with the component of its momentum opposite in sign from that of the beam particles and its average velocity over the trajectory will also be negative. Clearly increasing the beam radius beyond $R_A$ will not increase the current. If we estimate the average magnetic field as $\approx 1/2$ the field at the edge of the beam, we find that there is a maximum current which can be carried by a beam which satisfies our original assumptions:

$$\bar{B} \simeq \frac{I_A}{R_A} = \frac{I_A}{v_g} \simeq \frac{I_A}{\frac{pc}{q\bar{B}}} \quad . \tag{2.7}$$

Therefore

$$I_A \simeq \frac{pc}{q} = \frac{mc^2\gamma\beta}{q} \quad , \tag{2.8}$$

here $\beta = v/c$ and $\gamma = (1-\beta^2)^{1/2}$ . $I_A$ is called the "Alfven current limit" or the "Alfven-Lawson current limit" and for electrons we find

$$I_A \simeq 1700\ \beta\gamma \ , \tag{2.9}$$

in agreement with Alfven's (1939) more rigorous derivation. This restriction is much more stringent than the objections to the stored magnetic energy. For an electron energy of 100 keV, the currents estimated for the hypothetical small and large events are $\approx 10^{10}$ $I_A$ and $\approx 10^{13}$ $I_A$ respectively. The value of the current limit derived by Alfven depends on all the original assumptions being satisfied. Arbitrarily large currents can in principle be propagated by relaxing the assumption of exact electrostatic neutralization (Lawson 1957, 1958, 1959), the assumption that the beam is mono-energetic (Bennett 1934), the assumption that the current density (particle flux) is uniform (Hammer and Rostoker 1970) or adding a very strong magnetic field along the symmetry axis

(Hammer and Rostoker 1970). However, none of these mechanisms seem particularly likely to be applicable in solar flare impulsive hard X-ray bursts, although some are relevant to particular laboratory experiments. The simplest resolution to these objections is the existence of a reverse current (cf. 2.2).

## 2.2 Previous Work on Reverse Currents

It is well known that a plasma tends to preserve charge neutrality. A process which tends to give an excess positive or negative charge in some region will lead to electric fields which act upon the plasma. Movement of electrons in response to this electric field will then restore charge neutrality. One expects that analogous process will also tend to maintain current neutrality. If an electron beam is suddenly introduced into a plasma, a sudden change occurs in the magnetic field structure which will develop induced electric fields opposing the primary current.

Although interest in beams of relativistic electrons is not recent (see for example Bennett 1934, Alfven 1939), theoretical and experimental work on high current relativistic electron beams was stimulated by the development of devices capable of producing relativistic electron beams with currents on the order of or greater than the Alfven-Lawson current limit (See for example Graybill and Nablo 1966, Roberts and Bennett 1968, Yonas and Spence 1969). Roberts and Bennett (1968) injected a beam of 3.5 mev electrons ($\beta$=.992, $\gamma$=7.85) with a beam current of 3000 emu ($I \simeq .23 I_A$) into a linear pinch with $n_e \simeq 10^{18.5} cm^{-3}$. They found that the beam current was nearly completely neutralized by a reverse current in the ambient plasma and that the change in the total current (measured) was a very small fraction of the beam current. Similar results have been obtained with other experimental apparatus (Prono et al. 1975, Ekdahl et al. 1974, Goldenbaum et al. 1974, Klok et al. 1974, Miller and Kuswa 1973, Levine et al. 1971) when the ambient plasma density was sufficiently high.

Several theoretical models of energetic electron beams neutralized or partially neutralized by reverse currents in the ambient plasma have been developed (for example Cox and Bennett 1970, Hammer and Rostoker 1970, Lee and Sudan 1971, Lovelace and Sudan 1971, Chu and Rostoker 1973). Since these theoretical treatments are primarily concerned with the high current energetic electron beams that are typically produced in laboratory studies and not in the electron beams thought to be responsible for impulsive hard X-ray bursts, some of the results are not relevant to the solar flare case (cf. 2.3). The models cited treat cylindrically symmetric mono-energetic beams of the type considered by Alfven (cf. 2.1) with the possible addition of a uniform magnetic field along the symmetry axis. When the beam current is small compared to $I_A$, then the induced reverse current flows primarily outside the beam cylinder $(r > R)$ while for $I \gg I_A$ the reverse current is confined to $r \leq R$ and the current neutralization is local in the sense that the ambient electrons drift with the velocity

$$V_d = -\frac{n_b}{n_e} V_b \quad , \tag{2.10}$$

where $V_d$ is the reverse current drift velocity, $V_b$ is the velocity of the beam electrons and $n_b$ and $n_e$ are the beam and plasma electron number densities (Cox and Bennett 1970). Depending upon the sharpness of the leading edge of the beam, large amplitude coherent plasma oscillations may be generated by the passage of the beam head (Hammer and Rostoker 1970, Cox and Bennett 1970, Lee and Sudan 1971, Chu and Rostoker 1973). The amplitude of these plasma oscillations is $\sim(\omega_p T)^{-1}$, where $\omega_p$ is the plasma frequency $\left(\omega_p=(4n_e\pi\, e^2/m_e)^{1/2}\right)$ and $T$ is the rise time of the beam (Lee and Sudan 1971). These oscillations decay with a scale length of $V_b\,\tau_{ee}$, where $\tau_{ee}$ is a phenomenological momentum relaxation time for the plasma electrons. If the lateral dimension (R) of the beam is large compared to the electromagnetic skin depth $(\lambda_E=c/\omega_p)$, after the plasma oscillations decay the net current will be $\sim\lambda_E/R$ times the beam current. The current of the beam will be neutralized for a length of $\sim V_b\tau_{ee}(R/\lambda_E)^2$. The theoretical models for mono-energetic beams are not appropriate for the streams of energetic electrons that are responsible for impulsive X-ray bursts. We argue in Section 2.3 that the beams in solar flares will be current neutralized in steady state.

## 2.3 Steady State Model

We now examine a simple model for an impulsive X-ray burst. We consider a vertical flux tube extending from the corona to the chromosphere and assume that electrons are accelerated at the top of the flux tube by the development of stochastic electric fields (Sturrock 1966, Hall and Sturrock 1967, Newman 1973) or by some other mechanism (cf. Section 1.3). The injection of these electrons down the field toward the chromosphere then leads to the development of a reverse current both by the mechanisms considered for mono-energetic beams in laboratory plasmas (Cox and Bennett 1970, Hammer and Rostoker 1970, Chu and Rostoker 1973) and due to an electrostatic field due to charge imbalances. The strong tendency of a plasma to remain charge neutral implies that, if a current is generated in the plasma that would systematically violate $\partial\rho/\partial t=0$ on time scales much greater than a plasma period (i.e. a non-MHD current), then this current will generate a neutralizing secondary reverse current.

In contrast to the mono-energetic beams typical of laboratory experiments, the streams of energetic electrons that produce impulsive X-ray bursts probably have smooth distributions in energy. This is inferred from observations (cf. 1.2) and theoretical considerations indicate it is likely that the number of electrons does not increase with energy (Brown and Melrose 1977, Smith 1975). We consider below an energetic electron stream with a distribution of this type, that has electrons of all energies present. The low energy electrons are constantly merging with the background plasma and can build up charge imbalances. In the case of a mono-energetic beam considered by other workers (for example Cox and Bennett 1970, Chu and Rostoker 1973), charge imbalance would only build up at the ends of the plasma device since the energetic electrons do not interact with the plasma significantly except through the reverse current. Charge built up at the ends of an experimental plasma column would either be conducted away by external return paths or be shielded from the bulk of the plasma within a few Debye lengths of the ends and not drive reverse currents in most of the volume of the plasma column.

Lovelace and Sudan (1971) pointed out that the microscopic process involved in heating the plasma with reverse currents are equivalent to heating with currents induced by external fields. However, the reverse currents avoid the skin effect limitations of currents induced by external fields. Similarly, since charge can be supplied by the beam in the case of solar flares, charge imbalances can build up within the plasma and drive reverse currents. Although these charge imbalances arise throughout the plasma, we can estimate the time ($\tau_c$) required to accumulate sufficient charge separation from the time required to accumulate enough charge per unit area on a parallel plate capacitor to produce an electric field sufficient to drive the required reverse current. This required charge separation is related to the current density by

$$j\eta = E \simeq 4\pi\,\Sigma = 4\pi j_{unn}\, c\, \tau_c \, , \tag{2.11}$$

where $\Sigma$ is a surface charge density, $\eta$ is the resistivity, and $j_{unn}$ is the unneutralized portion of the beam current density. Then the time to accumulate the required charge is

$$\tau_c = \frac{j}{j_{unn}} \frac{\eta}{4\pi c} \quad . \tag{2.12}$$

The ratio of unneutralized current density to the beam current density is $\lambda_E/R$ (cf. 2.2) so that

$$\tau_c = \frac{\eta\, \omega_p R}{4\pi c^2} \quad , \tag{2.13}$$

$$\tau_c \simeq 10^{-9.16}(10^6/T)(n/10^9)^{1/2}(R/10^9) \; . \tag{2.14}$$

This assumes that the resistivity is the usual Spitzer value. If the resistivity is "anomalous" the effective collision frequency can be of order the electron plasma frequency ($\omega_p$) . Actually this is an upper limit, for the Buneman instability the effective collision frequency is

$\sim .1\omega_p$ (Buneman 1958). The resistivity is proportional to the collision frequency so we can write

$$\frac{\eta_{anom}}{\eta_{Spitzer}} \simeq \frac{10^{4.75} n^{1/2}}{10^{1.87} nT^{-3/2}} = 10^{7.38} (T/10^6)^{3/2} (10^9/n)^{1/2} \quad , \qquad (2.15)$$

so that $\tau_c$ becomes

$$\tau_c \simeq 10^{-1.78} (R/10^9) \ . \qquad (2.16)$$

We see that the time to accumulate charge imbalances sufficient to drive a neutralizing reverse current is short compared to time scales of interest.

If the resistivity is written

$$\eta = \frac{m_e c}{n_e e^2} \frac{1}{\tau_{ee}} \quad , \qquad (2.17)$$

then $\tau_c$ becomes

$$\tau_c = \frac{R}{c} (\omega_p \tau_{ee})^{-1} \ . \qquad (2.18)$$

and the ratio of the charge accumulation time to the time the current remains neutralized $(\tau_n)$ by the mechanisms considered for a mono-energetic beam (cf. 2.2) becomes

$$\frac{\tau_c}{\tau_n} = \frac{\frac{R}{c} (\omega_p \tau_{ee})^{-1}}{\frac{R}{c} \frac{R}{\lambda_E} (\omega_p \tau_{ee})} = (\omega_p \lambda_{ee})^{-2} (\lambda_E/R) \quad , \qquad (2.19)$$

$$\frac{\tau_c}{\tau_n} = 10^{-7.77} (\omega_p \tau_{ee})^{-2} (10^9/R)(10^9/n)^{1/2} \quad , \qquad (2.20)$$

so that the charge accumulation time is much shorter that the time the current would remain neutralized if no charge imbalance arose in the plasma. For time scales and length scales of interest in solar impulsive X-ray bursts, the reverse currents will be caused primarily by charge separation (Hoyng and Melrose 1977). Also, since beams of interest for solar impulsive X-ray bursts are not expected to have sharp fronts, the plasma oscillations excited by passage of the "beam head" will be of extremely small amplitude and consequently of no great significance (Melrose 1974). Therefore, we are justified in considering a steady state in which the beam current is exactly balanced by a reverse current in the background plasma. For the present (cf. Chapter 3), we assume that the background plasma can be adequately described by a Maxwellian velocity distribution and use transport coefficients based on this assumption (Sptizer 1962).

We are interested in the case in which the primary electron stream is composed of high-energy electrons with consequently long mean free paths in the tenuous solar corona. However, we shall find that the electric field that develops to drive the reverse current also decelerates the electron stream (cf. Lovelace and Sudan 1971). But when the electron energy becomes comparable with the thermal energy, the mean free path will be sufficiently short that the primary electrons will merge with the background plasma. As a simple representation of this process, we ignore collisions in discussing the primary beam but we assume that an electron of the primary beam is absorbed into the background plasma when it is decelerated to zero energy. This approximation is justified, if the temperature of the ambient plasma is sufficiently low.

If, as a further simplification, we consider a flux tube of uniform cross section, we may use the following simple one-dimensional form of the Vlasov equation:

$$v \frac{\partial f}{\partial s} + \frac{e}{m} \frac{d\phi}{ds} \frac{\partial f}{\partial v} = 0 \,, \tag{2.21}$$

where $s$ measures length along the tube, $v$ is velocity (along the tube), $f(s,v)$ is the velocity distribution function of the primary electron stream, and $\phi$ is the electrostatic potential.

At the top of the flux tube $(s=0)$, the primary electron stream is moving with positive velocity and electrons that are decelerated to zero velocity are assumed to be removed from the beam. Hence we may without ambiguity, express $f$ in terms of $\Psi$, which is defined by

$$\Psi = \frac{mv^2}{2e} \,. \tag{2.22}$$

The initial distribution function may therefore be expressed as

$$f(0,v) = F(\Psi) \,. \tag{2.23}$$

With this initial condition, we find that the solution of the Vlasov equation (2.21) is

$$f(s,v) = F(\Psi - \phi) \,. \tag{2.24}$$

The current density $j_s$ in the primary electron stream is given by

$$j_s = -\frac{e}{c} \int_0^\infty f(s,v) v \, dv \,, \tag{2.25}$$

which may be expressed as

$$j_s = -\frac{e^2}{mc}\int_0^\infty F(\Psi-\phi)d\Psi \ . \tag{2.26}$$

Since $\phi$ will prove to be negative in the region of interest, it is convenient to write

$$\Theta = -\phi \ , \tag{2.27}$$

so that Equation (2.26) may be reexpressed as

$$j_s = -\frac{e^2}{mc}\int_\Theta^\infty F(x)dx \ . \tag{2.28}$$

We have seen that the beam current will be nearly completely neutralized by currents in the background plasma, so we may write

$$j_p + j_s = 0 \ , \tag{2.29}$$

where $j_p$ is the secondary current induced in the background plasma. We here assume that the density and temperature are such that $j_p$ may be represented by Ohm's law,

$$j_p = \eta^{-1}E = \eta^{-1}\frac{d\Theta}{ds} \ . \tag{2.30}$$

It is convenient to introduce a new independent variable $\xi$ to replace $s$ by the relationship

$$d\xi = \eta \, ds \tag{2.31}$$

Then, on substituting Equations (2.28) and (2.30) into Equation (2.29) and differentiating with respect to $\xi$, we obtain

$$\frac{d^2\theta}{d\xi^2} + \frac{e}{mc} F(\theta) \frac{d\theta}{d\xi} = 0 \; . \tag{2.32}$$

It is convenient to solve this equation for $\xi$ in terms of $\theta$,

$$\xi = X(\theta) \quad , \tag{2.33}$$

rather than vice versa. Equation (2.32) becomes

$$\frac{mc}{e^2}\left(\frac{dX}{d\theta}\right)^{-2} \frac{d^2X}{d\theta^2} = F(\theta) \quad , \tag{2.34}$$

which may be integrated once to give

$$\frac{mc}{e^2}\left[\left(\frac{d\theta}{d\xi}\right)_{\xi=0} - \left(\frac{dX}{d\theta}\right)^{-1}\right] = \int_0^{\theta} F(\theta')d\theta' \quad , \tag{2.35}$$

if we assume that $\theta = 0$ ($\phi=0$) and $X = 0$ ($\xi=0$) at $s = 0$. We find from Equations (2.28), (2.29) and (2.30) that

$$\frac{e^2}{mc}\left(\frac{d\theta}{d\xi}\right)_{\xi=0} = \int_0^{\infty} F(\theta')d\theta' \quad . \tag{2.36}$$

Hence Equation (2.35) becomes

$$\frac{dX}{d\theta} = \frac{mc}{e^2}\left[\int_{\theta}^{\infty} F(\theta')d\theta'\right]^{-1} \tag{2.37}$$

It is now convenient to introduce a specific form for $F(\Psi)$:

$$F(\Psi) = K(\Psi_0+\Psi)^{-\gamma} \quad . \tag{2.38}$$

This is a power-law distribution at high energy which flattens at low energy, with the "knee" characterized by $\Psi_0$ .

We introduce the symbol $H(\Psi,s)$ for the flux of electrons ($cm^{-2}\ s^{-1}$) of energy exceeding $e\Psi$ at the position $s$ :

$$H(\Psi,s) = \frac{e}{m}\int_{\Psi}^{\infty} F\Big(\Psi' + \theta(s)\Big)\,d\Psi' \quad . \tag{2.39}$$

If the initial flux is written as $H_0(\Psi)$ , we find that

$$H_0(\Psi) = \frac{eK}{(\gamma-1)m}\,(\Psi_0 + \Psi)^{-\gamma+1} \quad , \tag{2.40}$$

so that the total particle flux is given by

$$H_0(0) = \frac{eK}{(\gamma-1)m}\,\Psi_0^{\;-\gamma+1} \quad . \tag{2.41}$$

With the form of Equation (2.38) for $F(\Psi)$ , Equation (2.37) integrates to give

$$X(\theta) = \frac{\gamma-1}{\gamma}\;\frac{mc}{e^2K}\left[(\Psi_0 + \theta)^{\gamma} - \Psi_0^{\gamma}\right] . \tag{2.42}$$

We easily obtain from Equation (2.42) an expression for the (negative) electric potential $\theta$ in terms of the resistivity weighted distance measure $\xi$ :

$$\theta(\xi) = \left(\Psi_0^{\gamma} + \frac{\gamma}{\gamma-1}\;\frac{e^2K}{mc}\,\xi\right)^{1/\gamma} - \Psi_0 \quad . \tag{2.43}$$

Hence from Equation (2.39), we find that

$$H(\Psi, s) = \frac{eK}{(\gamma-1)m}\left[\Psi + \left(\Psi_0^{\gamma} + \frac{\gamma}{\gamma-1}\,\frac{e^2K}{mc}\right)^{1/\gamma}\right]^{-(\gamma-1)} . \qquad (2.44)$$

On noting that the electric current carried by the stream is related to $H(\Psi, s)$ by

$$j_s(s) = -\frac{e}{c}\,H(0, s) \quad , \qquad (2.45)$$

we see that

$$j_s(s) = -\frac{e^2K}{(\gamma-1)mc}\left(\Psi_0^{\gamma} + \frac{\gamma}{\gamma-1}\,\frac{e^2K}{mc}\,\xi\right)^{-[(\gamma-1)/\gamma]} . \qquad (2.46)$$

In order to specify the current, particle flux, and electric field as functions of $s$, we must adopt a specific form for $\eta(s)$. A convenient approximation to the density and temperature structure of the solar atmosphere, which is expressible in analytic form, is provided by the constant heat flux model. If we now assume that $s$ measures distance vertically downward from the corona, and that $n = n_0$ and $T = T_0$ at $s = 0$, this model (Adams and Sturrock 1975) yields the following expressions:

$$T(s) = (T_0^{7/2} - bF\,s)^{2/7} \quad , \qquad (2.47)$$

$$n(s) = n_0[T_0/T(s)]\,\exp\left\{-\left[(T_0^{7/2} - bFs)^{5/7} - T_0^{5/2}\right]\right\}, \qquad (2.48)$$

where $a \simeq 10^{-1.21}$, $b \simeq 10^{6.58}$, and $F$ (ergs cm$^{-2}$ s$^{-1}$) is the downward heat flux.

The resistivity, in modified Gaussian units, may be derived from the expression given by Spitzer (1962):

$$\eta = gT^{-3/2} \tag{2.49}$$

where $g \simeq 10^{3.64}$ . Hence we find from Equation (2.31) that $\xi$ is related to $s$ by

$$\xi = \frac{7}{4}\frac{g}{bF}\left[T_0^2 - \left(T_0^{7/2} - bFs\right)^{4/7}\right] \tag{2.50}$$

Our model is then completely specified by the choice of the coronal temperature, the coronal density, the coronal heat flux, $\gamma$ , the energy corresponding to $\Psi_0$ , and the injected energetic electron flux. For the coronal parameters, we adopt values typical of the corona above an active region (Noyes 1971):

$$T \simeq 3 \times 10^6 \text{ K} \quad ,$$

$$n \simeq 10^9 \text{ cm}^{-3} \quad ,$$

$$F \simeq 5 \times 10^6 \text{ erg cm}^{-2} \text{ s}^{-1} \quad .$$

We choose $\Psi_0$ to correspond to 25 keV; and we choose $\gamma = 2.5$ . The fraction of the beam energy deposited and the total energy deposited by Joule heating between $T = 3 \times 10^6$ K and $T = 3 \times 10^4$ K as a function of the energetic electron flux are displayed in Figure 2.1. For a flare area of $10^{19.5}$ cm$^2$ , the energetic electron flux inferred from a large impulsive X-ray burst corresponds to $\sim 10^{17}$ cm$^{-2}$ s$^{-1}$ (Hoyng et al. 1976). Figure 2.2 illustrates the energy deposition rate due to Joule heating as a function of temperature of the atmosphere for this injected energetic electron flux. The ordinate of Figure 2.2 is the time required to raise the ambient plasma temperature by $10^7$ K, if the plasma were heated at the steady state rate. As we will see (cf. Chapter 3), the heating rate

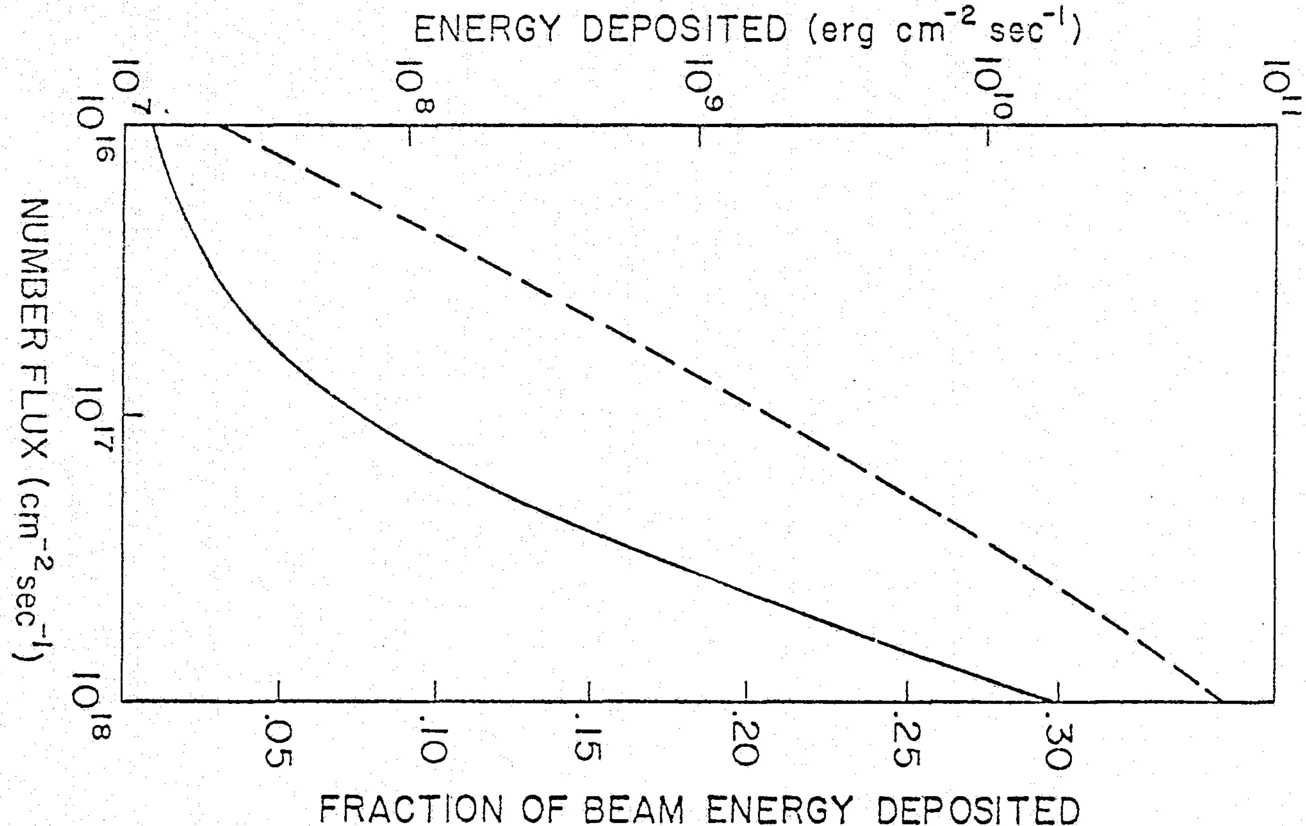


Figure 2.1 The fraction of beam energy deposited (solid curve) and the total energy deposited (broken curve) by Joule heating as a function of the energetic particle number flux.

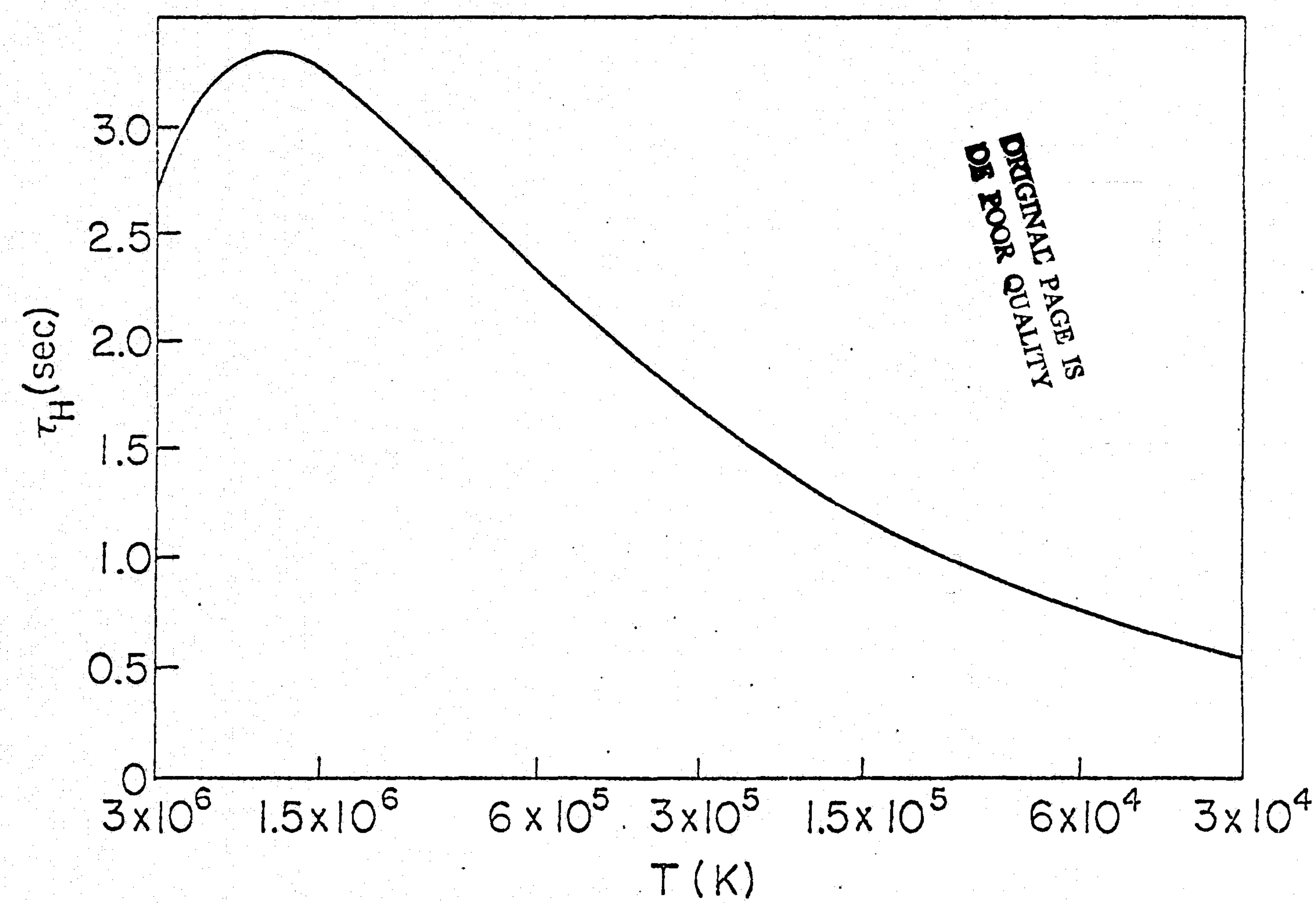




Figure 2.2 Joule heating rate as a function of ambient plasma temperature for an energetic particle number flux of $10^{17}$ cm$^{-2}$ s$^{-1}$. The heating rate is displayed as the time $\tau_H$ required to heat the plasma by $10^7$K .

decreases as the temperature of the plasma increases, so the ordinate of Figure 2.2 is only representative of the heating rate immediately after the beam is turned on.

We can check the assumption that Coulomb collisions are not important for the energetic electrons in the beam. Since the reverse current losses are proportional to the resistivity for a constant current density, these losses are proportional to $T^{-3/2}$. The Coulomb collisional losses for energetic electrons of the same kinetic energy are proportional to $n$. Therefore, we expect the ratio of reverse current losses to collisional losses to be proportional to $(nT^{3/2})^{-1}$. Reverse current losses will be more important than Coulomb collisional losses for an energetic electron in the beam if

$$\alpha \equiv 10^{-.35}(V/V_t)^2(V_d/V_t) > 1 \,, \qquad (2.51)$$

where $V$ is the velocity of the energetic electron, $V_t$ is the electron thermal velocity, and $V_d$ (cf. Equation 2.10) is the reverse current drift velocity (Hoyng et al. 1978). As we expected, for the same kinetic energy and current density, $\alpha$ is proportional to $(nT^{3/2})^{-1}$ since $V_t \propto T^{1/2}$ and $V_d \propto n^{-1}$. If we define the injected energetic electron flux as the flux of electrons with kinetic energies greater than $e\Psi_0$ then from Equation (2.40) we find

$$H_E = \frac{eK}{(\gamma-1)m2^{\gamma-1}}(\Psi_0)^{-\gamma+1} \quad . \qquad (2.52)$$

Since the reverse current drift velocity is related to the current density by

$$V_d = \frac{c\, j_p}{en} \quad , \tag{2.53}$$

we may write the drift velocity in terms of the injected energy flux as

$$V_d = \frac{2^{\gamma-1}}{n} H_E \left(1 + \frac{\gamma 2^{\gamma-1} e}{c\Psi_0} H_E\, \xi\right)^{-[(\gamma-1)/\gamma]} \quad . \tag{2.54}$$

For the adopted values $\gamma=2.5$ and $e\Psi_0=25$ keV, we find that the ratio of reverse current losses to Coulomb collisional losses $(\alpha)$ for a beam electron with kinetic energy 25 keV in the adopted constant heat flux model atmosphere is

$$\alpha = 10^{2.82} \frac{H_E}{nT^{3/2}} \left[1 + 10^{-23.56} \frac{H_E}{F} \; T_0^2 - T^2\right]^{-[(\gamma-1)/\gamma]} \tag{2.55}$$

In Figure 2.3, the energy at which $\alpha=1$ is plotted as a function of temperature for several values of $H_E$. We see that for any energetic electron flux we have considered, the energy at which Coulomb collisions are as important as the reverse current losses for the energetic electrons is reasonably low, indicating that our assumption that Coulomb collisions may be neglected is an adequate approximation.

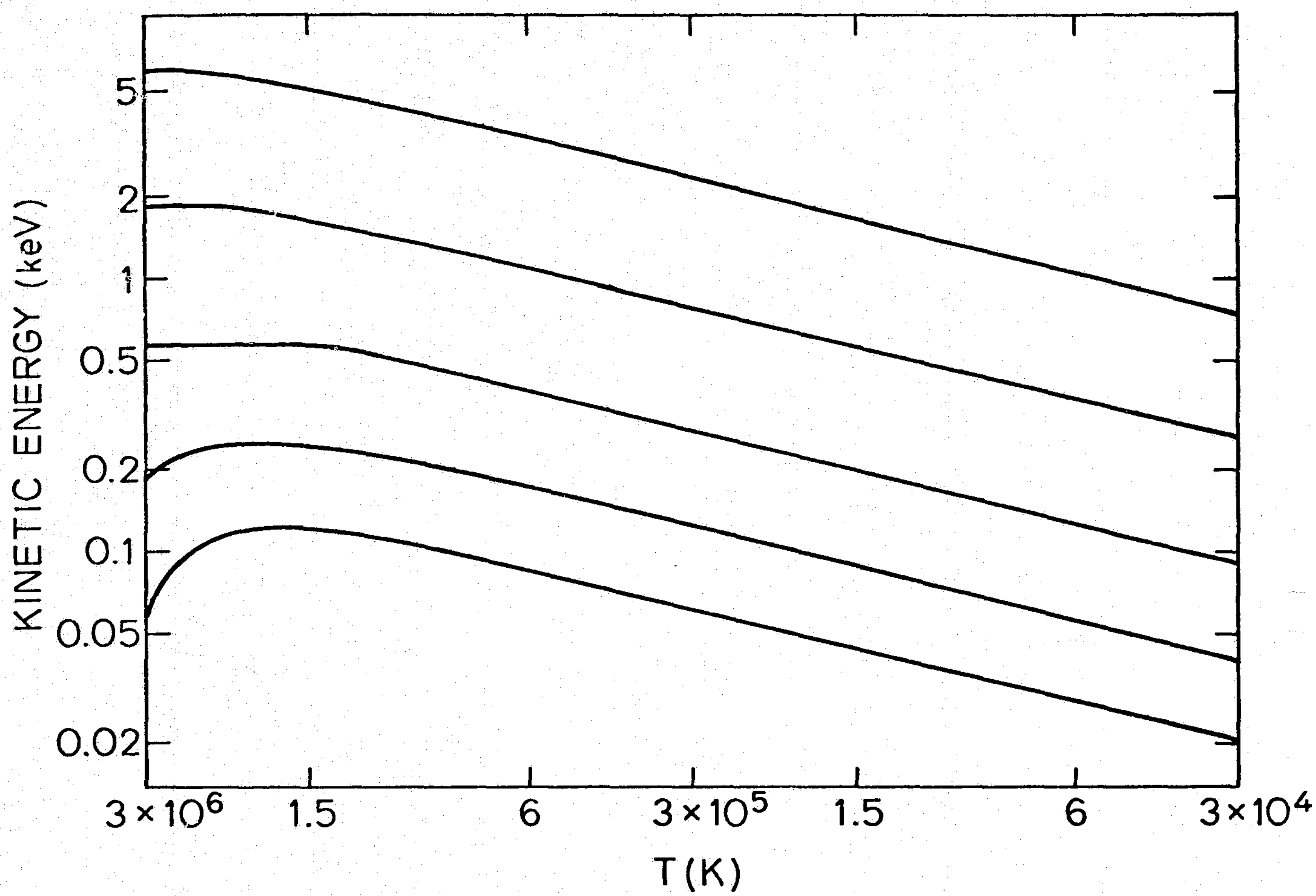


Figure 2.3 The kinetic energy for which the reverse current losses are equal to Coulomb collisional losses for a beam electron as a function of temperature in the constant heat flux atmosphere, for five different energetic particle number fluxes. From top to bottom the curves correspond to energetic particle number fluxes ($cm^{-2}s^{-1}$) of $10^{16.5}$, $10^{17.0}$, $10^{17.5}$, $10^{18.0}$ and $10^{18.5}$.

## 3. REVERSE CURRENT HEATING

### 3.1 Generalization of Steady Model to Time Dependent Case

As we have indicated, for the energetic electron fluxes required to account for the observed X-ray flux by thick-target bremsstrahlung, the ambient plasma is rapidly heated by the reverse current. The rate at which the background plasma is heated by the reverse current depends on the beam current density and the ambient plasma density and temperature. If the ratio of $V_d$ to the electron thermal velocity $(V_{t,e} = (2kT_e/m_e)^{1/2})$ is large enough, the background plasma may be unstable to the growth of electrostatic plasma turbulence which can dramatically enhance the plasma resistivity and therefore, the reverse current heating rate. For example, the reverse current will be unstable against the excitation of ion-acoustic or electrostatic ion-cyclotron turbulence unless for $T_e$ and $T_i$ the electron and ion temperatures, respectively, (Kindel and Kennel 1971)

$$V_d/V_{t,e} \le \begin{cases} 2.5 & \text{for } T_e \simeq .1\, T_i \text{ (ion-acoustic turbulence)} \\ .9 & \text{for } T_e \simeq .3\, T_i \text{ (ion-cyclotron turbulence)} \\ .3 & \text{for } T_e \simeq T_i \text{ (ion-cyclotron turbulence)} \\ .1 & \text{for } T_e \simeq 3\, T_i \text{ (ion-cyclotron turbulence)} \\ .05 & \text{for } T_e \simeq 10\, T_i \text{ (ion-acoustic turbulence)} \end{cases}$$

Reverse current heating is a self quenching process. If the reverse current is stable against the growth of electrostatic turbulence, then, as the plasma is heated by the reverse current, the resistivity decreases and the reverse current losses are reduced. If the reverse current is unstable to the growth of electrostatic turbulence, the plasma will be heated until the instability criterion is no longer satisfied. The heating of the plasma will also cause a pressure imbalance. The time $\tau$ (s) for the plasma to respond to changes of pressure by bulk motions can be estimated from

$$\tau \simeq L/V_{t,i} \quad , \tag{3.1}$$

where $L$ is a characteristic length and $V_{t,i}$ is the ion thermal velocity. Even for a temperature as large as $10^7$ K, this time is long ($10^2$ s) compared with the heating time for a length scale of $10^{9.7}$ cm, so that the plasma density will not change appreciably during the heating. Since we expect reverse currents to be established locally on time scales on the order of a plasma period ($\leq 10^{-9}$ s) which is much shorter than the time scale for heating of the plasma ($\geq 10^{-2}$ s), it should be a reasonable approximation to use the results of Chapter 2 for the instantaneous velocity distribution of the energetic electrons as a function of distance from the injection point.

We have calculated the heating due to reverse currents for two injected energetic electron fluxes ($H_E$). The heating rate was taken to be just that which results from the Ohmic losses suffered by the reverse current and is given by

$$\frac{\partial T}{\partial t} = \frac{c}{3n_e K} \eta j_p^2 \quad . \tag{3.2}$$

The electron and ion temperatures were assumed to be equal. We shall say more about this assumption later. At each time step the current at each spatial grid point was calculated using Equation (2.46). The time step was regulated so that the largest change in temperature at any grid point was 1% in one time step. Since we have not found an analytic solution to the time dependent problem considered here, the constant heat flux model of the atmosphere was abandoned in favor of a more accurate numerical model which is discussed in Appendix B. The spatial grid spacing was chosen so that for the initial temperature profile ($t=0$) the temperature change between spatial grid points was less than 1%. The atmosphere was assumed static; that is, the number density ($n$) was held constant in time. The details of the numerical methods used are discussed in Appendix A.

We have used the same $\gamma$ and $\Psi_0$ as in Chapter 2. The results for an injected energetic electron flux of $1.414 \times 10^{17}$ are displayed in Figure 3.1 while similar curves for an injected energetic electron flux of $5.656 \times 10^{17}$ are displayed in Figure 3.2. Figure 3.3 depicts the density structure of the model atmosphere. The abscissa, $I$, of the figures is integrated number density from the injection point, defined by

$$I(s) = \int_0^s n(s')ds' \quad , \tag{3.3}$$

where $n$ is the total number density (sum of neutral hydrogen and proton density). Because we have used a numerical model rather than the simple analytic constant heat flux model, we were not free to choose the density at the injection point (see Appendix B). The initial density in the adopted model is approximately twice the density in the constant heat flux model used in Chapter 2. Since the reverse current heating rate is proportional to $j_p^2$ and inversely proportional to density, the

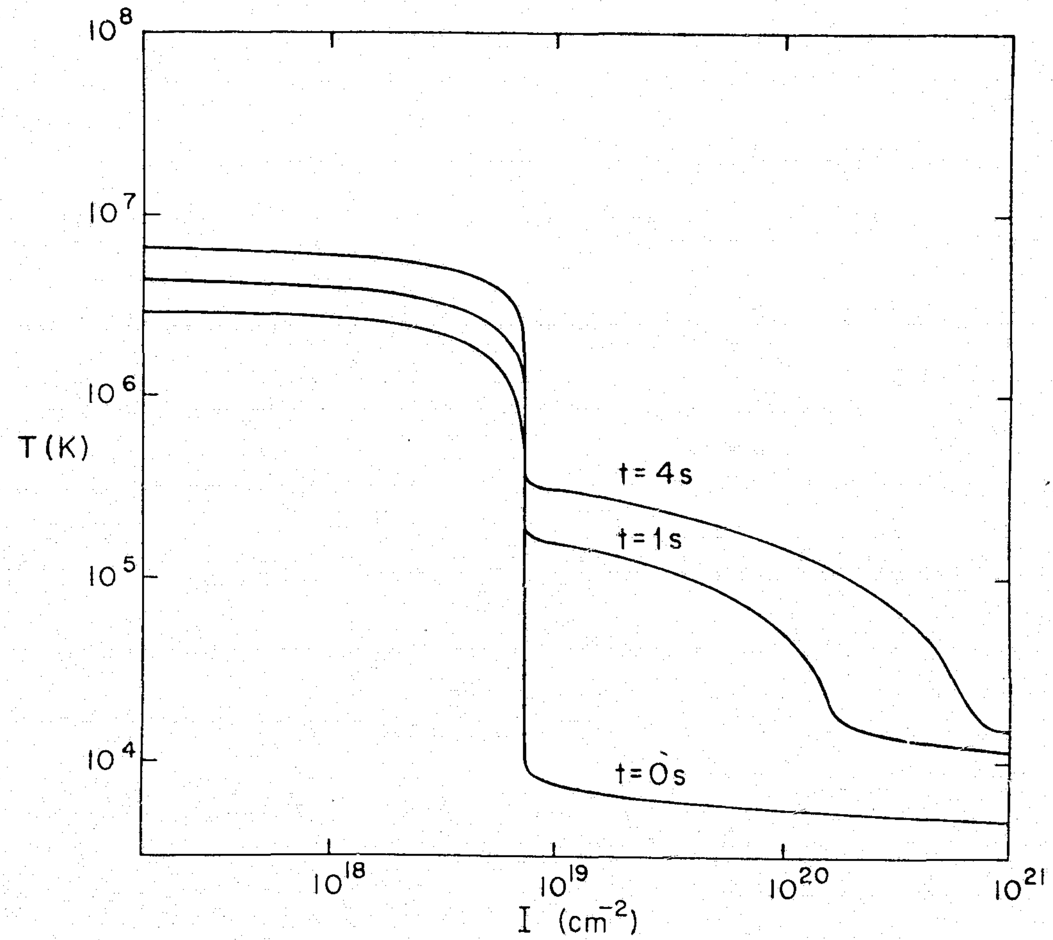


Figure 3.1 Temperature (T) as a function of integrated number density (I) from the injection point, for an energetic electron number flux of $1.414 \times 10^{17}$ cm$^{-2}$ s$^{-1}$. The temperature is displayed before the beam is injected (t = 0 s) and for two times after the beam is injected (t = 1 s and t = 4 s).

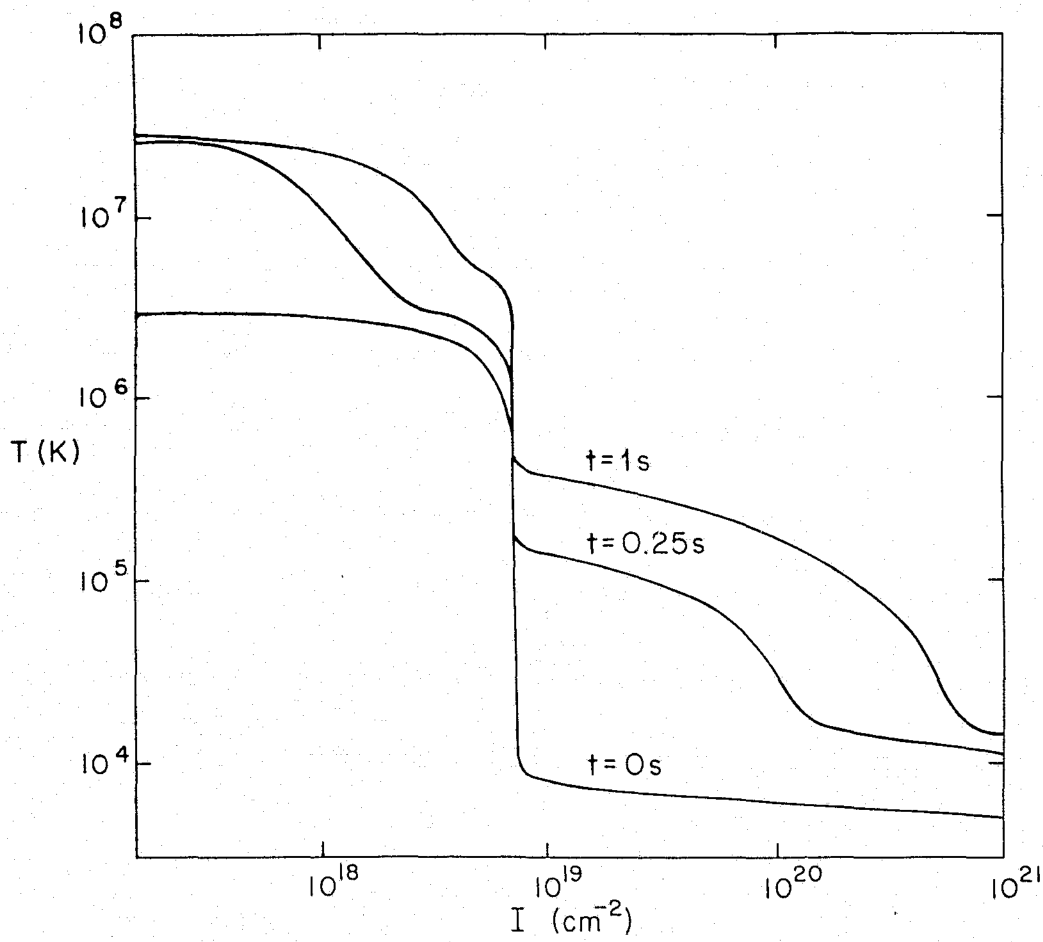


Figure 3.2 Temperature (T) as a function of integrated number density (I) from the injection point, for an energetic electron number flux of $5.656 \times 10^{17}$ cm$^{-2}$ s$^{-1}$. The temperature is displayed before the beam is injected (t = 0.25 s and t = 1 s).

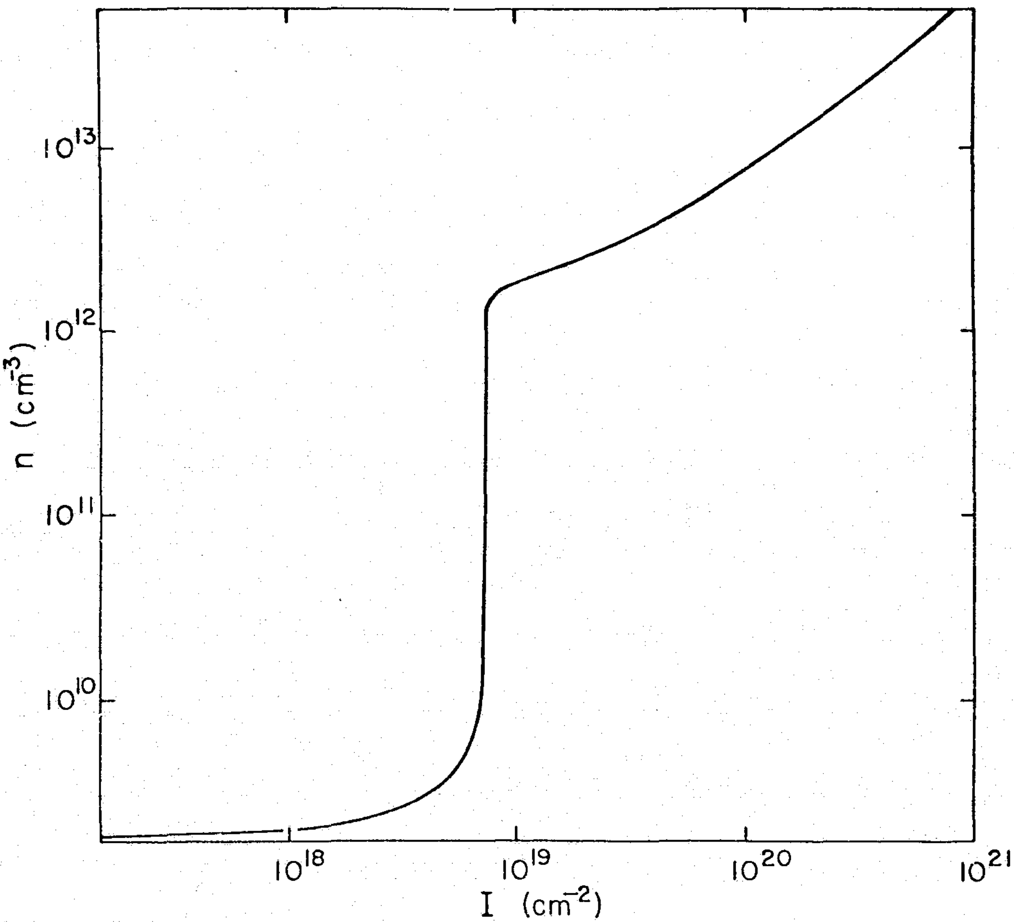


Figure 3.3 Number density of neutral and ionized hydrogen (n) as a function of integrated number density (I) from the injection point. The model serves only to represent the gross overall structure of the solar atmosphere above an active region (see Appendix B).

lower energetic electron flux corresponds roughly to the initial heating rate shown in Figure 2.2.

Figure 3.1 shows the temperature as a function of I for two times, 1s and 4s after the injection of the beam. The energetic electron flux used in the calculation of the results displayed in Figure 3.2 is four times that used for Figure 3.1. Figure 3.2 displays the temperature after .25s and 1s corresponding to the same total energy input as the curves for 1s and 4s in Figure 3.1. Thermal conductivity was neglected in these calculations, but computer runs with thermal conductivity included indicated that thermal conductivity did not have significant effects for the short time scales ($\leq$ 4s) involved in here (see Appendix A).

## 3.2 Anomalous Resistivity and Reverse Current Heating Rate

The electrical resistivity used in the calculations depended on the reverse current drift velocity as indicated below:

$$\eta = \begin{cases} \eta_s & V_d \leq 13\ V_{t,i} \\ \eta_s + \eta_A & V_d \geq 13\ V_{t,i} \end{cases} , \qquad (3.4)$$

where $V_{t,i} = (2KT_i/m_i)^{1/2}$ for $T_i$ the ion temperature, $m_i$ is the proton mass, $\eta_s$ is the resistivity due to Coulomb collisions derived by Spitzer (1962), and $\eta_A$ is an anomalous resistivity due to the presence of electrostatic ion-cyclotron turbulence calculated by Ionson (1976). For the purposes of calculating the value of the anomalous resistivity we have adopted B=100 gauss, a reasonable value for the pre-flare corona. Since we are considering a flux tube of constant cross section, the field is the same for all values of s , the distance from the injection point. For the smaller energetic electron flux, the reverse current drift velocity did not exceed the critical velocity for the onset of electrostatic ion-cyclotron turbulence. In this case the temperature of the tenuous coronal plasma was raised by a factor of $\approx 2$, but most of the beam energy was deposited in the dense portion of the model atmosphere.

The larger energetic electron flux, however, caused the reverse current drift velocity to exceed the critical velocity for the onset of electrostatic ion-cyclotron turbulence in the low density portion of the atmosphere, resulting in an anomalous resistivity and an order of magnitude increase in the temperature in these regions in a relatively short time. Since Coulomb collisions were neglected in this calculation, the heating of the denser portion of the atmosphere is not calculated accurately after the first few tenths of a second (also see Appendix B). If collisions were taken into account for the primary electrons in the beam, the heating of the denser regions below the corona would be more localized and higher temperatures would be reached. However, these results indicate that an energetic electron beam may significantly heat the low density coronal plasma much more rapidly than would be calculated from considering only the effects of Coulomb collisions on the beam electrons.

The time for electron and ion temperatures to equilibrate by Coulomb collisions assuming only one species rather than both species are heated as we have assumed may be estimated (Spitzer 1962) from

$$t_{ei} \approx 12.6\ n^{-1} \left(T_e + \frac{m_e}{m_i}\ T_i\right)^{3/2} \quad s \ . \qquad (3.5)$$

For the temperatures, densities, and the time scales considered here, Coulomb collisions alone will not establish equal electron and ion temperatures. We have taken the electron and ion temperatures to be equal for computational convenience; however, and must, therefore, address the question of whether one species is preferentially heated.

For the case depicted in Figure 3.1, for which the resistivity is just classical Spitzer resistivity, only the electrons are heated at first. According to Equation (3.5), the ions are not likely to be heated significantly in turn by energy exchange with the electrons. The heat capacity of the plasma is therefore reduced by a factor of 2, and the times given in Figure 3.1 should simply be reduced by a factor of 2.

The situation in which plasma turbulence develops, as for the case depicted in Figure 3.2, is considerably more complicated. As we have indicated, the critical drift velocity for the onset of electrostatic ion acoustic or ion-cyclotron turbulence depends on the ratio of the electron and ion temperatures. Just what happens when this drift velocity is exceeded is not well understood, however.

The anomalous resistivity which we have assumed to result from the presence of electrostatic ion-cyclotron turbulence was calculated by Ionson (1976) under the assumption that the turbulence saturates by ion resonance broadening (Dum and Dupree 1970) and that the electron and ion temperatures were equal. Palmadesso et al. (1974), on the other hand, made the first of these assumptions and calculated heating rates of electrons and ions. They found that the ions are heated much more rapidly than the electrons, and Papadopoulos (1977) has subsequently concluded that the instability turns off when the ion heating has proceeded to the point at which the instability criterion is no longer satisfied. If only the ions are heated, the situation will differ from that depicted in Figure 3.2. The temperature plotted should be interpreted as the ion temperature (note that this affects the calculation of the expected excitation and ionization rates) and the times given reduced by a factor of 2 for the same reason those in Figure 3.1 should be reduced if only the electrons are heated.

It has also been suggested that the ion-cyclotron turbulence saturates, not by ion resonance broadening, but by the formation of a plateau on the electron velocity distribution, instead, in which case no significant anomalous resistivity results (Papadopoulos 1977). If this happens, then as in the case without plasma turbulence, only the electrons are heated at first, at a rate given approximately by classical resistivity. In this case, however, larger electron beam current densities must have been involved to begin with in order for the reverse current drift velocity to have exceeded the critical velocity for the onset of ion-cyclotron turbulence. Since $j$ is larger than in the case without turbulence, the classical heating rate is higher for this case. If the electrons are heated sufficiently in this manner, the critical drift velocity for the onset of ion-acoustic turbulence will be exceeded. In that case, the electrons will be heated until the criterion for instability is no longer satisfied, or until

$$V_d \simeq C_s \simeq (kT_e/m_i)^{1/2} \quad , \tag{3.6}$$

where $C_s$ is the ion sound speed.

This latter scenario for rapid electron heating would apply, for instance, to an electron beam strength equal to that assumed in Figure 3.2. More precisely, assuming the ion-cyclotron turbulence does saturate by electron plateau formation, a beam of this strength would result first in electron heating given approximately by the results in Figure 3.1 with a time scale reduced by a factor of about 32. After roughly .5 s, ion-acoustic turbulence would develop, resulting in rapid heating of the electrons to a final temperature which may be estimated from

$$T_e \simeq m_i V_d^2/k \simeq 10^{10}\ \mathrm{K} \quad . \tag{3.7}$$

In short, the exact behavior of the ratio of the electron and ion temperatures is not well understood and cannot be determined without a much more detailed analysis than is appropriate for the present work. We have assumed that the electron and ion temperatures are about equal as a useful and reasonable approximation with which to estimate the magnitude of the reverse current heating. As discussed above, however, temperature enhancements much larger than those depicted in Figures 3.1 and 3.2 are possible.

# 4. CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

We have examined a simple model for the production of impulsive hard X-ray bursts during solar flares. The model involves a beam of energetic electrons propagating from the corona to the chromosphere. We have found that if this beam is to exist, the current carried by the beam electrons must be neutralized by a reverse current in the background plasma. The requirement that the reverse current exist has two consequences that have not been previously recognized in the context of this type of simple model of impulsive hard X-ray bursts. The reverse current heats the ambient plasma and the electric field that is developed to drive the reverse current decelerates the primary electrons. Joule heating by the reverse current is a more effective mechanism for heating the tenuous coronal plasma than Coulomb collisional losses from the energetic electrons, because the ohmic losses are caused by thermal electrons in the reverse current which have much shorter mean free paths than do the energetic electrons.

We have found that the time scale for heating the ambient plasma by reverse currents can be comparable with the time scales characteristic of impulsive X-ray bursts (Hoyng et al. 1976). It is possible that thermal bremsstrahlung from the rapidly heated plasma can account for a significant portion of the observed impulsive X-ray flux. Hence this mechanism can offer an explanation of the fact that some flares first produce high-energy X-ray emission near the top of a loop rather than at the footpoints of the loop (Brueckner 1976). Another important consequence of this process is that, if thermal emission can account for a substantial fraction of the impulsive flux up to $\sim 50$ keV, then the number of electrons required to produce the nonthermal X-ray flux is greatly reduced (Brown 1975).

The time scale for heating can also be short compared to the ionization times of the plasma ions and may therefore produce non-equilibrium line-emission strength enhancements of lines present in the plasma spectrum just prior to the rapid heating (Shapiro and Knight 1978). These non-equilibrium effects are likely to be observable only if plasma turbulence develops causing a large enhancement in the plasma resistivity (Shapiro and Knight 1978).

We have made several simplifying assumptions in order to facilitate the calculations presented in Chapters 2 and 3. In a more realistic model, some or perhaps all of these restrictions could be relaxed. We now briefly discuss how the relaxation of some of these assumptions is likely to change the conclusions we have drawn and suggest possible extensions of the work we have presented in Chapter 3. We have assumed that all the electrons in the beam are moving in the same direction, or equivalently that they have zero pitch angle. The reverse current arises to balance the flux of electrons in a given direction due to any anisotropy in the energetic electron velocity distribution. If the energetic electron velocity distribution is nearly isotropic, no significant reverse current will arise (see for example Smith and Lilliequist 1978). Even if the distribution is strongly anisotropic, but the electrons streaming down from the corona to the chromosphere have non-zero pitch angles, the Coulomb collisional losses will be enhanced relative to reverse current losses since the collisional losses are proportional to the total path length of the energetic electrons in the atmosphere, while the reverse current losses are proportional to the average component of the energetic electron velocity along the field. Since the emergent X-ray spectrum is relatively insensitive to the angular distribution of the energetic electron velocities (Langer and Petrosian 1977), it is extremely difficult to infer the reverse current drift velocity from measurements of the X-ray flux. A more detailed discussion of this and other difficulties in inferring the reverse current drift velocity from X-ray observations can be found elsewhere (Hoyng et al. 1978).

We have neglected the effects of Coulomb collisions on the primary electrons. As Figure 2.3 demonstrates, this is an adequate approximation immediately after the flux of energetic electrons is initiated; however, Coulomb collisions become relatively more important as the plasma is heated since the reverse current losses are reduced. Until a significant increase in the density of the coronal plasma is effected by the evaporation of material from the chromosphere, Coulomb collisions are unlikely to be important in the upper portions of the atmosphere. In the lower lying dense regions, Coulomb collisions will rapidly dominate over reverse current losses, and, as we have indicated, affect the heating of this portion of the atmosphere. One extension of the work presented in Chapter 3 that should provide additional insight into the behavior of energetic electrons in the solar atmosphere during flares would be to perform a calculation similar to that we have presented, but include the effects of Coulomb collisions and a distribution of pitch angles for the energetic electrons.

We have neglected the dynamics of the background plasma. As we have indicated, the rapid heating of the plasma can cause a large pressure imbalance. For the results presented in Figures 3.1 and 3.2, the pressure is a factor of $\sim 20$ higher in the high density portion of the atmosphere than in the low density regions indicating that evaporation of high density material would occur if the dynamics of the ambient plasma were accounted for. This would not have a large effect on the calculations presented in Chapter 3 because the time scales considered are so short. However, on longer time scales mass motions in the atmosphere could have important effects. Previous work with fluid dynamic models of solar flares (for example see Kostyuk and Pikel'ner 1975, Kostyuk 1975, Craig and McClymont 1976) has not included reverse current losses. The development of a numerical fluid dynamic model of the solar atmosphere heated by a beam of energetic electrons, including reverse current losses could provide valuable information about the formation of the quasi-thermal soft X-ray plasma that is produced during solar flares.

We have not calculated either the radiation from the heated plasma or the bremmstrahlung from the energetic electrons. Since almost all the information we now have and are likely to accumulate in the foreseeable future about solar flares comes from the observation of the emitted radiation, it would be useful to calculate the emitted radiation from any realistic model to ascertain to what degree it resembles the solar atmosphere during a flare.

More realistic models than those we have considered that include the effects of Coulomb collisions, the dynamics of the background plasma, a reasonable magnetic field configuration, radiation and thermal conduction are necessary to account for the complicated phenomena that are observed in solar flares. However, our study of the reverse current and the heating it can cause indicates that reverse currents can play an important role, at least in the initial heating of the solar plasma during a flare.

## Appendix A

## NUMERICAL AND COMPUTATIONAL METHODS

As we have indicated in Chapter 3, the results for the time dependent case are calculated by using the steady state results for the current as a function of distance from the injection point and calculating the change in temperature from a suitably discretized form of equation (3.2). In reality, the calculation is done for the more general case of partially ionized hydrogen. Since the reverse current heating calculation is only accurate in the tenuous high temperature portion of the atmosphere, this generalization did not have a substantial effect on the results of the calculation. However, the manner in which the partial ionization is included could in principle be accurate in any astrophysical plasma that is sufficiently tenuous that the gas is optically thin to its own radiation, photo-excitation and ionization are unimportant and collisional de-excitation can be ignored. The ionization state of the plasma is then a function of temperature only provided non-equilibrium effects can be ignored. The only elements in astrophysical plasmas that are sufficiently abundant for their ionization potential to affect the heat capacity of the gas are hydrogen and helium. Only hydrogen is included in the present calculation, but since the effects of partial ionization on the heat capacity are included via a pretabulated interpolation table (discussed below) the effects of helium could be included with only minor modification. The modified version of equation (3.2) actually solved numerically is

$$\frac{\partial T_E}{\partial t} = \frac{2c}{3nk} \eta \, j_p^2 \quad , \tag{A.1}$$

where $T_E$ is defined by

$$T_E = (1 + \chi)T + \chi \frac{2E_{ION}}{3k} \quad , \tag{A.2}$$

where $\chi(T)$ is the fraction of the hydrogen nuclei that are ionized and $E_{ION}$ is the ionization potential of hydrogen. That is, the thermal energy content of the plasma per cubic centimeter is

$$E_{TH} = \frac{3}{2} nk \, T_E \quad . \tag{A.3}$$

The temperature is obtained from $T$ via the interpolation tables mentioned above, and it is obvious that the inclusion of helium only involves calculating a different interpolation table. In fact we have included only hydrogen and used the expression given by Moore and Fung (1972) for $\chi(T)$ :

$$\chi(T) = \left(1 + 10^{-5.69}\beta \, e^{\beta}\left[.4288 + \frac{1}{2} \ln \beta + .4698\beta^{-1/3}\right]\right)^{-1} , \tag{A.4}$$

where $\beta$=15800/t. Then $T$ is implicitly defined as a function of $T_E$ by Equations (A.3) and (A.4).

Spitzer gives the resistivity of a hydrogen plasma as:

$$\eta_s = 10^{2.34} T^{-3/2} \ln \Lambda \; , \tag{A.5}$$

where $\Lambda$ is defined by

$$\Lambda = \begin{cases} 10^{4.09}\, T^{3/2} n_e^{-1/2} & T \le 4.2 \times 10^5 \mathrm{k} \\ 10^{4.09}\, T^{3/2} n_e^{-1/2} \left(\frac{4.2 \times 10^5}{T}\right)^{1/2} & T \ge 4.2 \times 10^5 \mathrm{k} \end{cases} \quad , \tag{A.6}$$

so that we may write $\eta_s$ as

$$\eta_s = \begin{cases} 10^{3.31}\, T^{-3/2} \left[3/2 \ln T - \frac{1}{2} \ln \chi - \frac{1}{2} \ln n\right] & T \le 4.2 \times 10^5 \mathrm{k} \\ 10^{3.54}\, T^{-3/2} \left[\ln T - \frac{1}{2} \ln \chi - \frac{1}{2} \ln n\right] & T \ge 4.2 \times 10^5 \mathrm{k} . \end{cases} \tag{A.7}$$

Therefore, $\eta_s$ can be written as a sum of a function of $T$ only and a function of $T$ only times $\ln(n)$ :

$$\eta_s = TL(T) + TM(T) \ln(n) \quad , \tag{A.8}$$

where

$$TL(T) = \begin{cases} 10^{3.31}\, T^{-3/2} \left[3/2 \ln T - \frac{1}{2} \ln \chi\right] & T \le 4.2 \times 10^5 \mathrm{k} \\ 10^{3.54}\, T^{-3/2} \left[\ln T - \frac{1}{2} \ln \chi\right] & T \ge 4.2 \times 10^5 \mathrm{k} \quad , \end{cases} \tag{A.9}$$

and

$$TM(T) = \begin{cases} 10^{3.01}\, T^{-3/2} & T \le 4.2 \times 10^5 \mathrm{k} \\ 10^{3.24}\, T^{-3/2} & T \ge 4.2 \times 10^5 \mathrm{k} \quad . \end{cases} \tag{A.10}$$

The calculation of the current as a function of distance depends only on the resistivity weighted distance measure $\xi$ . In Chapter 2 we were able to write an analytic expression for $\xi$ as a function of $s$ , but in the present case the resistivity varies with time. The value of $\xi$ at the ith grid point is approximated by

$$\xi_i^j = \xi_{i-1}^j + (\eta_{i-1}^j + \eta_i^j)\ (s_i - s_{i-1})/2 \quad , \tag{A.11}$$

where superscripts refer to time steps and subscripts refer to spatial grid points, and $\xi_1^j = 0$ . So long as the reverse current drift velocity is less than the critical velocity for the onset of ion cyclotron turbulence, the calculation of $\xi_i^j$ in this manner is straightforward. However, when the background plasma is unstable to the growth of ion cyclotron turbulence, the situation is somewhat more complicated. In this case the value of $\eta_i^j$ depends on the current, and a transcendental equation must be solved to find $\eta_i^j$ from Equations (2.46), (2.49) and the result for anomalous resistivity due to ion cyclotron turbulence (Ionson 1976)

$$\eta_a = 0.06\ (c\ \Omega_i/\omega_{pe}\ \omega_{pi})(1 - 13V_{t,i}/V_d) \quad , \tag{A.12}$$

where $\Omega_i = (eB/m_i c)$ and $\omega_{p_\alpha} = (4\pi n_\alpha e/m_\alpha)^{1/2}$ , we find that $J_i^j$ is defined implicitly by

$$J_i^j = \frac{e^2 K}{(\gamma-1)mc}\left\{\psi_0^\gamma + \frac{\gamma e^2 K}{(\gamma-1)mc}\left[\xi_{i-1}^j + (\eta_{i-1}^j + \eta_{s_i}^j)\ (s_i - s_{i-1})/2\right.\right.$$
$$\left.\left. + .06\left(\frac{m_e}{m_i}\right)^{1/2} \frac{B}{4\pi e}\ \frac{(s_i - s_{i-1})}{2n_i X_i^j}\left(1 - V_{c_i}^j/V_{d_i}^j\right)\right]\right\}^{[(\gamma-1)/\gamma]} \quad , \tag{A.13}$$

where $V^j_{d_i} = cJ^j_i/en\chi^j_i$ and $V^j_{c_i} = 13\ V_{t,i}$ . If we define $G(J^j_i)$ by

$$G(J^j_i) = J^j_i + \frac{e^2K}{(\gamma-1)mc}\left\{\psi_0^\gamma + \frac{\gamma e^2K}{(\gamma-1)mc}\left[\xi^j_{i-1} + (\eta^j_{i-1} + \eta^j_{s_i})\ (s_i - s_{i-1})/2\right.\right.$$

$$\left.\left. + .06\left(\frac{m_e}{m_i}\right)^{1/2} \frac{B}{4\pi e}\ \frac{(s_i - s_{i-1})}{2n_i\chi^j_i}\left(1 - \frac{V^j_{c_i}\ e\ n_i\chi^j_i}{cJ^j_i}\right)\right]\right\}^{[(\gamma-1)/\gamma]}, \tag{A.14}$$

then when $G(J^j_i)=0$ , $J^j_i$ satisfies Equation (A.13). When the current calculated neglecting anomalous resistivity corresponds to a drift velocity that is greater than $V^j_{c_i}$ , we take an initial estimate for $J^j_i$ , $J'^j_i$ :

$$J'^j_i = \frac{V^j_{c_i}}{c}\ e\ n_i\ \chi^j_i\ , \tag{A.15}$$

and refine this estimate by application of Newton's method to Equation (A.14). Examination of Equation (A.13) shows that Newton's method will always converge for this initial estimate and the convergence is usually reasonably rapid, i.e. usually 6 or fewer iterations are required.

The functions needed for the calculation [TL, TM, X and the implicitly defined $T(T_E)$] are evaluated by cubic interpolation on pretabulated tables. The method used is dependent on the architecture of the IBM 360-370 series computers and the internal representation of double precision floating point numbers used on these machines. The use of pretabulated functions is considerably faster than calls to the FORTRAN library routines that would otherwise be necessary. This is particularly true in the case of the implicitly defined function $T(T_E)$ which would have to be solved iteratively at each spatial grid point for each time step. The internal representation of double precision floating numbers on IBM 360-370 computers is presented diagramatically below.

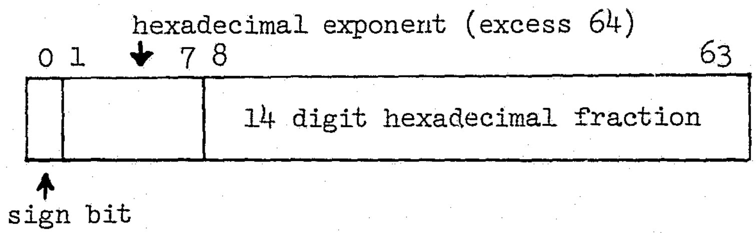


FIG. A.1. Internal representation of double precision floating point numbers on IBM 360 and 370 series computers.

In the interpolation procedure, the first 16 bits (bits 0-15) are extracted, an offset substracted and the result treated as a double word displacement from the base address of an interpolation table. The remaining 48 bits (16-63) are used to form a floating point fractional displacement (frac) from the largest value of the temperature for which the function is tabulated which is smaller than the value of the temperature for which the value of the function is desired. The value of this displacement is such that $0 \leq \text{frac} < 1$ ; frac is used to calculate weights for the four nearest tabulated values of the function in a cubic polynomial interpolation. Once the weights for the cubic interpolation are calculated only 4 double precision floating point multiplies ($\sim .61\ \mu s$ each on IBM 370-168 with high speed multiply) and 3 adds ($\sim .30\ \mu s$ each) are required to produce an interpolated value from a table. Since the weights are to be calculated for TL, TM and $\chi$ they are also used to calculate the critical velocity for the onset of ion-cyclotron turbulence. This would require a call to the FORTRAN library subroutine "DSQRT" which is sufficiently fast that interpolation would be slower for the calculation of the square root alone. However, since the weights must be calculated for TL, $\chi$ , TM and TE, interpolation is faster than a call to "DSQRT" because the weights are effectively "free" for this calculation. The semi-logarithmic tabulation scheme allows interpolation from $T=4.096 \times 10^3$ K to $6.710 \times 10^7$ K with a maximum relative displacement from a value of T for which the function is tabulated of $\sim 3\%$ with only 820 table entries. In fact some of the 820 entries are never used due to the nature of the signed magnitude normalized representation of floating point numbers on these machines, but the reason for using this sort of tabulation scheme is that a reasonably large range can be covered with relatively few table entries, and the correct tabulated values can be accessed extremely rapidly.

Listings of two main programs and several subroutines are provided for the sake of completeness. All of the time consuming routines have been hand coded in assembly language. Routines that perform initialization and diagnostic functions as well as the main program are coded in FORTRAN. The first main program produces the tables that are required for the cubic interpolation. The second main program reads in the interpolation tables, model parameters and starting values. The starting values used initially are from the steady state model atmosphere described briefly in Appendix B.

The assembly language programs calculate the current at each spatial grid point (CURCAL), calculate the change in temperature at each point and determine the time step (TESTP) and write out the arrays at the designated intervals (TOUT). In addition the calculation of the current

(CURCAL) requires taking the $-(\gamma-1)/\gamma$ power of a number, which if done with FORTRAN library routines would require taking the natural logarithm and exponentiating. Both the library routines "DLOG" and "DEXP" are slower than "DSQRT" so an assembly language routine was written that calculates the 3/5 power of a number (F35), the routine is called by CURCAL. The subroutine DIAG is used primarily for monitoring the performance of the model during program changes and subsequent debugging. In production runs it could be replaced with a subroutine that does nothing (i.e. returns as soon as it is called) without affecting the model calculations; therefore it is not reproduced here. The FORTRAN subprograms initialize the array containing $T_E$ (EINIT) and read in the starting values (INIT and RDR). The calculation that includes thermal conductivity which is referred to in Chapter 3 is not discussed in detail here. In order to avoid undue restriction on the time step [to satisfy the Courant-Friedrichs-Lewy condition - see Richtmyer and Morton (1967)], the method employed is implicit and requires the inversion of a tridiagonal matrix (dimension 846) and is rather slow. Runs with this program indicated thermal conduction did not change the results substantially so these routines are not reproduced here.

ORIGINAL PAGE IS
OF POOR QUALITY

TABULATION ROUTINE

```
      IMPLICIT REAL*8 (A-H,O-Z)
C
C     THIS PROGRAM CALCULATES SEMI=LOGARITHMIC TABLES NEEDED FOR
C     CALCULATION OF REVERSE CURRENT HEATING WITH RESISTIVITY THAT
C     IS A FUNCTION OF THE DRIFT VELOCITY.
C
C     THE TABLES ARE WRITTEN OUT TO LOGICAL UNIT 9
C
      REAL*8 TL(820),TM(820),VITH(820),CHIN(820),CINV(564)
      REAL*8 DTO/Z431000000000000/,TBO/Z441000000000000/,
     .KAY/1.38054D-16/,EC/4.80298D-10/,MH/1.67530890-24/,
     .PI/Z413243F6A8885A31/,MP/1.67252D-24/,ME/9.109D-28/,
     .HBAR/1.05450D-27/,MU,EION,ONE3/ZC055555555555555/,
     .C/2.997925D10/,CSIG,LSIG,K32,CKAP,LNS1,LNS2,DCON,BETA,PSI
      INTEGER*4 LIM(4)/242,242,242,51/
 9001 FORMAT(10A8)
C
C     CALCULATE CONSTANTS
C
      MU=(ME*MP)/(ME+MP)
      EION=(MU*EC**4)/(2.DO*HBAR**2)
      TION=(2.DO*EION)/(3.DO*KAY)
      CSIG=0.5D0*PI
      K32=KAY*DSQRT(KAY)
      CSIG=(0.5D0*CSIG*DSQRT(CSIG*ME)*EC*EC*C)/K32
      LSIG=(3.DO*K32)/(2.DO*EC*EC*EC*DSQRT(PI))
      CVITH=2.DO*169.DO*KAY/MP
      CCHIN=1.5*KAY
      LNS1=DLOG(LSIG)
      LNS2=DLOG(LSIG*4.2D5)
      DCON=1.DO/(4.5D5*1.58D5)
C
C     CALC TABLES:  ELECTRICAL CONDUCTIVITY (TL,TM), CRITICAL
C     VELOCITY (VITH*13.) AND INVERSE IONIZATION FRACTION (CHIN).
C
      DT=DTO
      TO=TBO
      DO 20 II=1,4
         K=(256*(II-1))+1
         L=K+LIM(II)
         T=TO-DT
         DO 10 J=K,L
            BETA=1.58D5/T
            PSI=4.5D5/(BETA*DEXP(BETA)*(.4288D0+.5D0*DLOG(BETA)
     .      + .4698D0*BETA**ONE3))
            VITH(J)=DSQRT(CVITH*T)
            CHIN(J)=CCHIN*(1.DO+PSI)/PSI
            CHI=PSI/(1.DO+PSI)
            IF(T.GT.4.2D5)GOTO 5
            TM(J)=CSIG/(T*DSQRT(T))
            TL(J)=TM(J)*(LNS1+1.5D0*DLOG(T)-.5D0*DLOG(CHI))
            GOTO 10
    5       TM(J)=CSIG/(T*DSQRT(T))
            TL(J)=TM(J)*(LNS2+DLOG(T)-.5D0*DLOG(CHI))
   10       T=T+DT
         TO=TO*16.DO
   20    DT=DT*16.DO
C
C     CALCULATE TABLES FOR CHI INVERSE ((1+CHI)*T+CHI*TION AS FUNC OF T)
C
      TNEW=TBO-DTO
      DT=DTO
      TO=TBO
      DO 40 II=1,3
         K=(256*(II-1))+1
```

```
      L=K+LIM(II+1)
      T=T0-DT
      DO 30 J=K,L
         TCH=4.D-16*TNEW
25          TOLD=TNEW
            BETA=1.58D5/TOLD
            EBETA=DEXP(BETA)
            B13=BETA**ONE3
            TEMP1=0.4288D0+0.5D0*DLOG(BETA)+.4698D0*B13
            PSI=4.5D5/(BETA*EBETA*TEMP1)
            C=PSI/(1.D0+PSI)
            DC=DCON*C*C*BETA*BETA*EBETA*((1.D0+BETA)*
     .      TEMP1 + (.5D0-.1566D0*B13))
            TNEW=TOLD-((1.D0+C)*TOLD+C*TION-T)/(1.D0+C+
     .      (TOLD+TION)*DC)
            IF(DABS(TNEW-TOLD).GT.TCH)GOTO 25
         CINV(J)=TNEW
30       T=T+DT
      T0=T0*16.D0
40    DT=DT*16.D0
C
C     WRITE OUT TABLES
C
      WRITE(9,9001)TL
      WRITE(9,9001)TM
      WRITE(9,9001)VITH
      WRITE(9,9001)CHIN
      WRITE(9,9001)CINV
      STOP
      END
```

REVERSE CURRENT HEATING MAIN ROUTINE

```
      IMPLICIT REAL*8 (A-H,O-Z)
C
C     THIS PROGRAM CALCULATES REVERSE CURRENT HEATING OF A MODEL
C     ATMOSPHERE READ IN AS UP TO 1024 VALUES OF TEMPERATURE (T) NUMBER
C     DENSITY (N) AND DISTANCE (S) FROM THE INJECTION POINT (TOP OF
C     MODEL.  THE PROGRAM DOES NOT HAVE TO START AT TIME 0 (INJECTION
C     TIME) AS THE CURRENT TIME, TIME STEP AND ITERATIONS TO THIS POINT
C     ARE READ IN ALSO.  THE PROGRAM READS IN THE MAXIMUM NUMBER OF
C     ITERATIONS TO BE PERFORMED (NITER), THE ENERGETIC ELECTRON NUMBER
C     FLUX (EFLUX), PSIO WHICH CORRESPONDS TO AN ENERGY CHARACTERISTIC
C     OF A LOW ENERGY KNEE IN THE ENERGETIC ELECTRON DISTRIBUTION
C     (SEE KNIGHT AND STURROCK 1977) , FRAC, THE MAXIMUM PERCENTAGE
C     CHANGE IN THE THERMAL ENERGY CONTENT OF THE PLASMA PER HYDROGEN
C     NUCLEUS ALLOWED AT ANY GRID POINT IN ONE TIME STEP, TIMMAX
C     THE MAXIMUM TIME FOR THIS RUN (REAL TIME NOT COMPUTER TIME) AND
C     DTOUT, THE INTERVAL AT WHICH THE ARRAYS CONTAINING THE TEMPERATURE
C     AND CURRENT DENSITY AS WELL AS THE CURRENT TIME AND TIME STEP.
C
C  CALLED SUBROUTINES:
C
C     INIT - READS IN STARTING VALUES OF DENSITY, TEMPERATURE AND
C            DISTANCE AS WELL AS TIME, TIME STEP AND NUMBER OR PREVIOUS
C            ITERATIONS.
C
C     EINIT - CALCULATES INITIAL TE DEFINED AS (1+CHI)T+2*EION/3*K FOR
C             EACH SPATIAL GRID POINT.  ENERGY INPUT INCREASES TE
C             AND T IS CALCULATED FROM CHINV.
C
C     NOUT - WRITES OUT DENSITY AND DISTANCE ARRAYS AS WELL AS INPUT
C            PARAMETERS (TO FORTRAN LOGICAL UNIT 9)
C
C     CURINT - INITIALIZATION FOR CURCAL (SEE CURCAL)
C
C     TESTPI - INITIALIZATION FOR TESTP (SEE TESTP)
C
C     CURCAL - CALCULATES CURRENT AS A FUNCTION OF DISTANCE USING
C              STEADY STATE RESULTS OF KNIGHT AND STURROCK AND A
C              RESISTIVITY THAT DEPENDS ON THE REVERSE CURRENT DRIFT
C              VELOCITY.
C
C     TESTP - UPDATES TEMPERATURE AND ADJUSTS TIME INCREMENT SO THAT
C             MAXIMUM CHANGE IN TE AT ONE GRID POINT IS FRAC*TE AT THE
C             GRID POINT
C
C     DIAG - OUTPUTS A SMALL SUBSET OF THE CALCULATED CURRENT DENSITY
C            AT INTERVALS DETERMINED BY VALUES IT READS FROM LOGICAL
C            UNIT 5 - CAN BE RECOMPILED WITHOUT RECOMPILING THE REST
C            OF THE PROGRAMS AS IT DOES NOT AFFECT CALCULATIONS.
C
C     TOUT - WRITES OUT TEMPERATURE AND CURRENT ARRAYS AND CURRENT
C            TIME, TIME STEP AND ITERATIONS TO LOGICAL UNIT 9
C
C     CTOUT - CLOSES LOGICAL UNIT 9 (I.E. END FILE 9)
C
C
C  DECLARE VARIABLES:
C
      REAL*8 KAY,ME,C,GAM,NORM,SFST,EFLUX,PSIO,DTOUT,
     .EL,EFACT,NEWDT
      REAL*8 T(1024),N(1024),J(1024),S(1024),OSIG(1024),LN(1024),
     .TE(1024),TUP(1024),SD(1024),
     .TL(820),TM(820),CNV(820),VITH(820),CINV(564)
      INTEGER*4 IND(2,1024)
 5001 FORMAT(5F7.0)
 5002 FORMAT(I7)
```

```
 9001 FORMAT(10A8)
C
C     INITIALIZE CONSTANTS:  BOLTZMANN'S CONSTANT, ELECTRON REST MASS,
C     ELECTRON CHARGE (ESU), # ERG/KEV, SPEED OF LIGHT
C
      KAY=1.38054D-16
      ME=9.1091D-27
      EL=4.80298D-10
      EFACT=1.60210D-9
      C=2.997925D10
C
C     READ IN MODEL PARAMETERS
C
      READ(5,5002)NITER
      READ(5,5001)EFLUX,PSIO,FRAC,TIMMAX,DTOUT
      READ(8,9001)TL
      READ(8,9001)TM
      READ(8,9001)VITH
      READ(8,9001)CNV
      READ(8,9001)CINV
C
C     UNITS OF INPUT PARAMETERS ARE:
C     EFLUX 1.D17 (CM**2 SEC)**-1, PSIO IN KEV, FRAC IN PERCENT,
C     TIMMAX IN SEC (MAXIMUM TIME),DTOUT IN SEC (OUTPUT INTERVALS)
C
C     SCALE INPUT VARIABLES
C
      EFLUX=EFLUX*1.D17
      PSIO=(PSIO*EFACT)/EL
      FRAC=FRAC*0.01D0
C
C     CALCULATE CONSTANTS FOR CALCULATION OF CURRENT
C
      TEMP1=PSIO+PSIO
      GAM=2.5D0
      GAMM1=1.5D0
      TEMP1=TEMP1*DSQRT(TEMP1)
      TEMP2=PSIO*DSQRT(PSIO)
      NORM=(EFLUX*GAMM1*ME*TEMP1)/EL
      TEMP1=-(EL*EL*NORM)/(GAMM1*ME*C)
      J(1)=TEMP1/TEMP2
      TEMP2=TEMP2*PSIO
      TEMP3=-GAM*TEMP1
C
C     INITIALIZE TIME AND ITERATIONS
C
      NEWDT=0.D0
      IITER=0
      TIM=0.D0
C
C
C     INITIALIZE TEMPERATURE AND DENSITY
C
      NTAB=1024
      CALL INIT(T,N,S,DELT,TIM,NIT,NTAB)
      OUTIME=DTOUT+TIM
      NITER=NIT+NITER
      IITER=NIT
      T(NTAB+1)=T(NTAB)
      S(NTAB+1)=S(NTAB)+(S(NTAB)-S(NTAB-1))
      CALL EINIT(T,TE,NTAB)
C
C     OUTPUT INPUT PARAMETERS AND INITIAL DENSITY AND DISTANCES
C
      CALL NOUT(EFLUX,PSIO,FRAC,TIMMAX,NTAB,N,S)
C
C     NEED 1/N IN LOOP SO WE CHANGE N TO 1/N AND CALCULATE
C     DIFFERENCES OF DISTANCES USED IN TIME STEP.
C
```

```
      SFST=-S(2)
      DO 10 I=1,NTAB
      SD(I)=(S(I)-SFST)*0.5D0
      SFST=S(I)
      LN(I)=0.5D0*DLOG(N(I))
   10 N(I)=2.D0/(3.D0*KAY*N(I))
      LN(NTAB+1)=LN(NTAB)
C
C     PASS ADDRESSES OF INTERPOLATION TABLES AND OTHER
C     CONSTANTS TO CURCAL AND TESTP
C
      CALL CURINT(TEMP1,TEMP2,TEMP3,GAM,TL,TM,VITH,CNV,
     .N,LN,SD,OSIG,J,T,NTAB)
      CALL TESTPI(T,TE,J,OSIG,N,TUP,DELT,FRAC,NTAB,CINV)
C
C     START TIME STEPPING LOOP
C
    1    CONTINUE
C
C        CALCULATE CURRENT AND RESISTIVITY AT EACH GRID POINT
C
         CALL CURCAL(OSIG,J,T,N,LN,NTAB)
C
C        CALCULATE ONE TIME STEP WORTH OF HEATING, UPDATE TEMPERATURE
C        AND ADJUST TIME STEP ACCORDING TO FRAC
C
         CALL TESTP(T,TE,J,OSIG,N,TUP,DELT,FRAC,NTAB,CINV)
C
C        WRITE OUT SOME STUFF TO MAKE SURE THINGS ARE WORKING RIGHT
C
         CALL DIAG(S,T,TE,J,OSIG,N,TUP,DELT,TIM,NTAB,IITER)
C
C        STEP TIME
C
         IITER=IITER+1
         IF(IITER.GT.NITER)GOTO 40
         TIM=TIM+DELT
         IF(TIM.LT.OUTIME)GOTO 1
         OUTIME=OUTIME+DTOUT
C
C        OUTPUT CURRENT VALUES OF TIM,TEMP AND J
C
         CALL TOUT (TIM,T,J,DELT,IITER,NTAB)
         IF(TIM.LT.TIMMAX)GOTO 1
   40 CALL CTOUT
      WRITE(6,*)IITER,DELT
      STOP
      END
      SUBROUTINE EINIT(T,TE,NTAB)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 T(NTAB),TE(NTAB)
      REAL*8 KAY/1.38054D-16/,EC/4.80298D-10/,MP/1.67252D-24/,
     .ME/9.1091D-28/,
     .HBAR/1.05450D-27/,ONE3/Z0555555555555555/,
     .BETA,CHI
      TION=(ME*MP)/(ME+MP)
      TION=(TION*EC**4)/(2.D0*HBAR**2)
      TION=(2.D0*TION)/(3.D0*KAY)
      DO 10 I=1,NTAB
         BETA=1.58D5/T(I)
         CHI=4.5D5/(BETA*DEXP(BETA)*(.4288D0+.5D0*DLOG(BETA)
     .   +.4698*BETA**ONE3))
         CHI=CHI/(1.D0+CHI)
   10    TE(I)=T(I)+(CHI*(T(I)+TION))
      RETURN
      END
      SUBROUTINE INIT(T,N,S,DELT,TIM,NIT,NTAB)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 T(1024),N(1024),S(1024)
```

```
1001 FORMAT(10A8)
     READ(10,1001)NTAB,NIT,DELT,TIM
     CALL RDR(T,N,S,NTAB)
     RETURN
     END
     SUBROUTINE RDR(T,N,S,NTAB)
     REAL*8 T(NTAB),S(NTAB),N(NTAB)
1001 FORMAT(10A8)
     READ(10,1001)N
     READ(10,1001)S
     READ(10,1001)T
     RETURN
     END
```

# CURCAL

```
CURCAL   CSECT
*
*        ROUGHLY EQUIVALENT TO THE FORTRAN CODE BELOW, EXCEPT
*        THE FUNCTIONS PASSED IN THE ARGUMENT LIST OF THE
*        FORTRAN ENTRY POINT CURINT (TL,TM,VTH,CNV) ARE IMPLEMENTED
*        IN LINE IN THE ASSEMBLY LANGUAGE VERSION AND A CALL TO
*        THE ASSEMBLY LANGUAGE VERSION SHOULD PASS THE ADDRESSES
*        OF SEMI-LOGARITHMIC TABLES (TL(820),TM(820),VTH(820),CNV(820))
*        VIA THE ENTRY POINT CURINT RATHER THAN FUNCTION NAMES.
*        CURCAL DOES THE EQUIVALENT OF A FORTRAN RETURN 1 WHEN A
*        VALUE OF T IS OUTSIDE THE TABULATED RANGE.  THERE IS NO
*        OBVIOUS WAY TO MAKE THIS APPARENT IN THE FORTRAN VERSION.
*        THE ARGUMENTS PASSED TO CURCAL ARE IGNORED AND
*        OBTAINED FROM LOCAL STORAGE WHERE CURINT PUT THEM.
*
*        NOTE THAT THIS MEANS CURINT MUST BE CALLED BEFORE THE
*        FIRST TIME CURCAL IS CALLED OR A REAL MESS WILL RESULT.
*
*
*      SUBROUTINE CURINT(TEMP1,TEMP2,TEMP3,GAM,TL,TM,VTH,CNV,
*     .N,LN,SD,OSIG,J,T,NTAB)
*
*      DECLARE VARIABLES
*
*      IMPLICIT REAL*8 (A-H,O-Z)
*      REAL*8 N(NTAB),LN(NTAB),SD(NTAB),T(NTAB),J(NTAB),OSIG(NTAB),
*     .TL,TM,VTH,CNV
*      REAL*8 TEMP1,TEMP2,TEMP3,CONAN,ESU,C,CONAN1,B,CHT,
*     .VC,VITH,TSI,JA,NCON,NCON1,ME,MI,PI,JCON,JC1
*      DATA C/2.997925D10/,ESU/4.80298D-10/,ME/9.109D-28/,
*     .MI/1.67252D-24/,PI/Z413243F6A8885A30/,ERR/1.D-6/
*
*      INITIALIZE CONSTANTS FOR CURCAL
*
*      INTEGER*4 COUNT/30/
*      B=1.D2
*      CONAN=6.D-2*DSQRT(ME/MI)*B/(4.D0*PI*ESU)
*      TCON=((GAM-1.D0)/GAM)
*      CONAN1=-TCON*TEMP3*CONAN*ESU/C
*      TCON=TCON*TEMP1
*      CE=-C/ESU
*      RETURN
*      ENTRY CURCAL(OSIG,J,T,N,LN,NTAB,*)
*
*      CALCULATE CURRENT AND RESISTIVITY (OSIG) FOR FIRST POINT
*
*      OSIG(1)=TL(T(1))+TM(T(1))*LN(1)
*      CHT=CNV(T(1))
*      VITH=VTH(T(1))
*      VD=J(1)*N(1)*CHT*CE
*      IF(VD.LT.VITH)GOTO 10
*
*      THE NEXT STATEMENT CALCULATES THE ANOMALOUS PART OF THE
*      RESISTIVITY IF THE DRIFT VELOCITY IS GREATER THAN THE
*      CRITICAL VELOCITY FOR THE ONSET OF TURBULENCE
*
*      OSIG(1)=OSIG(1)+CONAN*N(1)*CHT*(1.D0-VITH/VD)
*
*      INITIALIZE TSI TO ZERO
*
*   10 TSI=0.D0
*      DO 20 I=2,NTAB
*      OSIG(I)=TL(T(I))+TM(T(I))*LN(I)
*      TSI=TSI+(OSIG(I-1)+OSIG(I))*SD(I)
*      JCON=(TEMP2+TEMP3*TSI)
```

```
*       F35C1=1.D0/JCON
*       F35C2=F35(F35C1)
*       CHT=CNV(T(I))
*       VITH=VTH(T(I))
*       J(I)=TEMP1*F35C2
*       NCONO=N(I)*CHT
*       CJA=NCONO*SD(I)*CONAN
*       NCON=NCONO*CE
*       VD=JT*NCON
*       IF(VD.LT.VITH)GOTO 20
*
*       THIS SECTION CALCULATES ANOMALOUS PART OF RESISTIVITY USING
*       NEWTON'S METHOD FOR INTERIOR GRID POINTS - SKIP IF
*       DRIFT VELOCITY IS LESS THAN CRITICAL VELOCITY
*
*       JA=(12.D0/13.D0)*VITH/NCON
*       JCON=JCON+CJA*TEMP3
*       CJA=CJA*JA
*       NCON1=NCON*CONAN1
*       J(I)=JA*(1.D0+(J(I)-JA)/(NCON1*SD(I)*F35C1*J(I)+JA))
*       DO 15 K=1,COUNT
*       CJAT=CJA/J(I)
*       TC1=TCON-CJAT
*       JC1=JCON-CJAT
*       JT=J(I)*((TC1+TEMP1*JC1)/(J(I)*JC1*F35(JC1)+TC1))
*       IF(DABS((JT-J(I))/JT).LE.ERR)GOTO 16
*    15 J(I)=JT
*    16 TSIG=NCONO*CONAN*(1.D0-JA/J(I))
*       TSI=TSI+TSIG*SD(I)
*       OSIG(I)=OSIG(I)+TSIG
*    20 CONTINUE
*       RETURN
*       END
*
         USING *,15
         B     CFIRST         BRANCH AROUND NAME, OTHER ENTRY ETC.
         DC    X'06'
         DC    CL7'CURCAL '
         ENTRY CURINT
         USING *,15
CURINT   B     IFIRST         BRANCH AROUND NAME, SAVE AREA ETC.
         DC    X'06'
         DC    CL7'CURINT '
AREA     DS    18F
REG1     DC    AL4(ARGA)      R1 (ADDR ARG LIST)
REG2     DC    F'0'           R2 BASE T
REG3     DC    F'0'           R3 (BASE OSIG - 8 - BASE T)
REG4     DC    F'0'           R4 (BASE J - BASE T)
REG5     DC    F'0'           R5 (BASE N - BASE T)
REG6     DC    F'0'           R6 (BASE LN - BASE T)
REG7     DC    F'0'           R7 (BASE SD - BASE T)
EIGHT    DC    F'8'           R8 (INCREMENT - 8)
REG9     DC    F'0'           R9 (BASE T + 8*(NTAB-1) COMPARAND)
TLA      DC    F'0'           R10 (BASE TL TABLE CHANGES IN LOOP)
TMA      DC    F'0'           R11 (BASE TM TABLE CHANGES IN LOOP)
VTHA     DC    F'0'           BASE VTH TABLE
CNVA     DC    F'0'           BASE CNV TABLE
COUNT    DC    F'30'          MAX # OF NEWTON'S METHOD ITERATIONS
ARGA     DC    X'80'          ARGUMENT LIST (ONE LONG)
         DC    AL3(F35A)      ADDR ARGUMENT
TBND     DC    X'00000330'
TDISP    DC    X'00004410'
         CNOP  0,8            FORCE DOUBLE WORD ALIGNMENT
WM1      DC    D'0.'
WP1      DC    D'0.'
WM3      DC    D'0.'
WP3      DC    D'0.'
FLOAT    DC    X'4000000000000000'
CONAN    DC    D'0.'
```

```
CONAN1   DC    D'0.'
TEMP1    DC    D'0.'
TEMP2    DC    D'0.'
TEMP3    DC    D'0.'
GAM      DC    D'0.'
NCON     DC    D'0.'
NCONO    DC    D'0.'
TCON     DC    D'0.'
JCON     DC    D'0.'
CJA      DC    D'0.'
TC1      DC    D'0.'
JA       DC    D'0.'
TSI      DC    D'0.'
ERR      DC    D'1.E-6'        ERROR TOLERANCE FOR NEWTON'S METHOD
C        DC    D'2.997925E10'
ESU      DC    D'4.80298E-10'
ME       DC    D'9.1091E-28'
MI       DC    D'1.67252E-24'
B        DC    D'1.E2'
PI       DC    X'413243F6A8885A30'
CE       DC    D'0.'
F35C1    DC    D'0.'
F35C2    DC    D'0.'
F35A     DC    D'0.'
IFIRST   STM   14,12,12(13)    SAVE CALLING ROUTINE'S GPR'S
         LR    2,13            R2 <= ADDR OLD SAVE AREA
         LA    13,AREA         R13 <= ADDR NEW SAVE AREA
         DROP  15              R15 NO LONGER BASE REG
         USING AREA,13         R13 NEW BASE REG
         ST    2,4(13)         LINK SAVE AREAS
         ST    13,8(2)
         LM    2,9,0(1)        R2-R9 <= ADDR'S 1ST 8 ARG'S
         LD    0,0(5)          F0 <= GAM
         ST    6,TLA           TLA <= BASE TL TABLE
         LDR   2,0             F2 <= GAM
         ST    7,TMA           TMA <= BASE TM TABLE
         LD    4,0(4)          F4 <= TEMP3
         ST    8,VTHA          VTHA <= BASE VTH TABLE
         STD   4,TEMP3         TEMP3 (LOCAL) <= TEMP3
         ST    9,CNVA          CNVA <= BASE CNV TABLE
         LD    6,0(3)          F6 <= TEMP2
         LM    4,10,32(1)     R4-R10 ADDR'S REST OF ARG'S
         LD    4,0(2)          F4 <= TEMP1
         L     10,0(10)        R10 <= NTAB
         STD   6,TEMP2         TEMP2 (LOCAL) <= TEMP2
         SLA   10,3            R10 <= NTAB*8
         STD   4,TEMP1         TEMP1 (LOCAL) <= TEMP1
         S     10,EIGHT        R10 <= 8*(NTAB-1)
         SD    2,=D'1.'        F2 <= GAM-1.D0
         SR    4,9             R4 <= BASE N - BASE T
         DDR   2,0             F2 <= (GAM-1.D0)/GAM
         SR    5,9             R5 <= BASE LN - BASE T
         STD   2,GAM           GAM <= (GAM-1.D0)/GAM
         SR    6,9             R6 <= BASE SD - BASE T
         LD    0,ME            F0 <= ME
         L     1,REG1          R1 <= ADDR ARG LIST
         DD    0,MI            F0 <= ME/MI
         L     15,=V(DSQRT)    R15 <= ENTRY ADDR DSQRT
         STD   0,F35A          F35A      <= ME/MI
         BALR  14,15           F0 <= DSQRT(ME/MI)
         SR    8,9             R8 <= BASE J - BASE T
         MD    0,B             F0 <= DSQRT(ME/MI)*B
         SR    7,9             R7 <= BASE OSIG - BASE T
         MD    0,=D'6.E-2'     F0 <= 6.D-2*DSQRT(ME/MI)*B
         AR    10,9            R10 <= BASE T + 8*(NTAB-1)
         LD    2,=D'4.'        F2 <= 4.D0
         MD    2,PI            F2 <= 4.D0*PI
         S     7,EIGHT         R7 <= BASE OSIG - 8 - BASE T
         MD    2,ESU           F2 <= 4.D0*PI*ESU
```

```
         ST    9,REG2           REG2 <= BASE T
         DDR   0,2              F2 <= 6.D-2*DSQRT(ME/MI)*B/(4.DO*PI*ESU)
         ST    7,REG3           REG3 <= BASE OSIG - 8 - BASE T
         STD   0,CONAN         CONAN <= 6.D-2*DSQRT(ME/MI)*B/(4.DO*PI*ESU)
         ST    8,REG4           REG4 <= BASE J - BASE T
         LD    4,GAM            F4 <= ((GAM-1.DO)/GAM)
         MD    0,TEMP3          FO <= TEMP3*CONAN
         ST    4,REG5           REG5 <= BASE N - BASE T
         MDR   0,4              FO <= ((GAM-1.DO)/GAM)*TEMP3*CONAN
         ST    5,REG6           REG6 <= BASE LN - BASE T
         LD    2,C              F2 <= C
         ST    6,REG7           REG7 <= BASE SD - BASE T
         LCDR  2,2              F2 <= -C
         DDR   0,2              FO <= -((GAM-1.DO)/GAM)*TEMP3*CONAN/C
         ST    10,REG9          REG9 <= BASE T + 8*(NTAB-1) COMPARAND
         MD    4,TEMP1          F4 <= TEMP1*((GAM-1)/GAM) = TCON
         DD    2,ESU            F2 <= -C/ESU
         STD   4,TCON           TCON <= TEMP1*((GAM-1)/GAM)
         L     10,4(13)         R10 <= ADDR OLD SAVE AREA
         MD    0,ESU            FO <= -((GAM-1.DO)/GAM)*TEMP3*CONAN*ESU/C
         STD   2,CE             CE <= -C/ESU
         LM    14,10,12(10)     GPR'S RESTORED
         STD   0,CONAN1         CONAN1 <= ((GAM-1.DO)/GAM)*TEMP3*CONAN
         L     13,4(13)         R13 <= ADDR OLD SAVE AREA
         SR    15,15            R15 <= 0 (RETURN CODE)
         MVI   12(13),X'FF'     INDICATE CONTROL RETURNED
         BR    14               RETURN
         DROP  13
         USING CURCAL,15
CFIRST   STM   14,12,12(13)     SAVE CALLING ROUTINE'S GPR'S
         LR    2,13             R2 <= ADDR OLD SAVE AREA
         LA    13,AREA          R13 <= ADDR NEW SAVE AREA
         DROP  15               R15 NO LONGER BASE REG
         USING AREA,13          R13 NEW BASE REG
         ST    2,4(13)          LINK SAVE AREAS
         ST    13,8(2)
         LM    1,11,REG1        SET UP GPR'S
         LH    12,0(2)          R12 <= HIGH ORDER 2 BYTES OF T(1)
         MVC   FLOAT+1(6),2(2) FLOAT <= FRACTIONAL DISPLACEMENT
         S     12,TDISP         REDUCE R12 BY TDISP. NOW # DOUBLE WORDS
*                               FROM BASE OF INTERPOLATION TABLES
         LD    4,FLOAT          F4 <= FRACTION 0 LE FRAC LT 1
         BM    BADT             IF RESULT NEGATIVE - OUT OF RANGE
*                               GOTO BADT
         C     12,TBND          IF R12 GREATER THAN TBND
         BH    BADT             GOTO BADT
         SD    4,=D'.5'
         SLA   12,3             R12 <= R2*8 NOW BYTE DISPLACEMENT
*                               FROM BASE OF INTERPOLATION TABLES
*
*        NOW COMPUTE WEIGHTS FOR CUBIC INTERPOALTION OF
*        FUNCTIONS OF T
*
         LDR   2,4              F2 <= X
         MDR   4,4              F4 <= X**2 = X2
         HDR   4,4              F4 <= X2/2
         SD    4,=D'1.125'      F4 <= X2/2 - 9/8
         LDR   6,4              F6 <= X2/2 - 9/8
         HDR   4,4              F4 <= X2/4 - 9/16
         MDR   6,2              F6 <= X3/2 - 9X/8
         LCDR  0,4              FO <= -X2/4 + 9/16
         ADR   0,6              FO <= X3/2 - X2/4 - 9X/8 + 9/16
         STD   0,WM1            WM1 <= WEIGHT FOR TABLE ENTRY CORRESPOND-
*                               ING TO CLOSEST SMALLER VALUE OF T.
         LCDR  0,4              FO <= -X2/4 + 9/16
         SDR   0,6              FO <= -X3/2 - X2/4 + 9X/8 + 9/16
         STD   0,WP1            WP1 <= WEIGHT FOR TABLE ENTRY CORRESPOND-
*                               ING TO CLOSEST LARGER VALUE OF T.
         ADR   6,2              F6 <= X3/2 - X/8
```

```
         AD     4,=D'.5'       F4 <= X2/4 - 1/16
         MD     6,=X'4055555555555555'  F6 <= X3/6 - X/24
         LDR    0,4            F0 <= X2/4 - 1/16
         SDR    0,6            F0 <= -X3/6 + X2/4 + X/24 - 1/16
         ADR    6,4            F6 <= X3/6 + X2/4 - X/24 - 1/16
         STD    0,WM3          WM3 <= WEIGHT FOR SMALLEST VALUE OF T
         STD    6,WP3          WP3 <= WEIGHT FOR LARGEST VALUE OF T
*
* NOW CALCULATE INTERPOLATED VALUES OF TL AND TM
* (HAVE WEIGHTS FOR TABLES ENTRIES 1 & 4 IN FPR'S 0 & 6)
*
         LDR    4,0            F4 <= WEIGHT 1
         LDR    2,6            F2 <= WEIGHT 4
         MD     0,0(10,12)     F0 <= WEIGHT 1 * TL 1
         MD     2,24(10,12)    F2 <= WEIGHT 4 * TL 4
         MD     4,0(11,12)     F4 <= WEIGHT 1 * TM 1
         MD     6,24(11,12)    F6 <= WEIGHT 4 * TM 4
         ADR    0,2            F0 <= W1*TL1 + W4*TL4
         ADR    4,6            F4 <= W1*TM1 + W4*TM4
         LD     2,WM1          F2 <= WEIGHT 2
         LDR    6,2            F6 <= WEIGHT 2
         MD     2,8(10,12)     F2 <= W2*TL2
         MD     6,8(11,12)     F6 <= W2*TM2
         ADR    0,2            F0 <= W1*TL1 + W4*TL4 + W2*TL2
         ADR    4,6            F4 <= W1*TM1 + W4*TM4 + W2*TM2
         LD     2,WP1          F2 <= WEIGHT 3
         LDR    6,2            F6 <= WEIGHT 3
         MD     2,16(10,12)    F2 <= W3*TL3
         MD     6,16(11,12)    F6 <= W3*TM3
         ADR    0,2            F0 <= INTERPOLATED VALUE OF TL
         ADR    4,6            F4 <= INTERPOLATED VALUE OF TM
         MD     4,0(6,2)       F4 <= TM*LN(1)
         ADR    4,0            F4 <= TL + TM*LN(1)
         L      10,CNVA        R10 <= BASE ADDR CHINV TABLE
         STD    4,8(3,2)       OSIG(1) <= TL + TM*LN(1)
         L      11,VTHA        R11 <= BASE VTH TABLE
         LD     0,WM3          F0 <= WEIGHT FOR SMALLEST VALUE OF T
         LD     6,WP3          F6 <= WEIGHT FOR LARGEST VALUE OF T
*
* NOW CALCULATE INTERPOLATED VALUES OF CH AND VT
* (HAVE WEIGHTS FOR TABLES ENTRIES 1 & 4 IN FPR'S 0 & 6)
*
         LDR    4,0            F4 <= WEIGHT 1
         LDR    2,6            F2 <= WEIGHT 4
         MD     0,0(10,12)     F0 <= WEIGHT 1 * CH 1
         MD     2,24(10,12)    F2 <= WEIGHT 4 * CH 4
         MD     4,0(11,12)     F4 <= WEIGHT 1 * VT 1
         MD     6,24(11,12)    F6 <= WEIGHT 4 * VT 4
         ADR    0,2            F0 <= W1*CH1 + W4*CH4
         ADR    4,6            F4 <= W1*VT1 + W4*VT4
         LD     2,WM1          F2 <= WEIGHT 2
         LDR    6,2            F6 <= WEIGHT 2
         MD     2,8(10,12)     F2 <= W2*CH2
         MD     6,8(11,12)     F6 <= W2*VT2
         ADR    0,2            F0 <= W1*CH1 + W4*CH4 + W2*CH2
         ADR    4,6            F4 <= W1*VT1 + W4*VT4 + W2*VT2
         LD     2,WP1          F2 <= WEIGHT 3
         LDR    6,2            F6 <= WEIGHT 3
         MD     2,16(10,12)    F2 <= W3*CH3
         MD     6,16(11,12)    F6 <= W3*VT3
         ADR    0,2            F0 <= INTERPOLATED VALUE OF CH
         ADR    4,6            F4 <= INTERPOLATED VALUE OF VT
         LDR    2,0            F2 <= CHT
         SDR    6,6            F6 <= 0.D0
         MD     2,CE           F2 <= CHT*CE
         L      10,TLA         R10 <= BASE TL TABLE
         MD     2,0(5,2)       F2 <= N(1)*CHT*CE
         L      11,TMA         R11 <= BASE TM TABLE
         STD    6,TSI          TSI <= 0.D0
```

```
         MD     2,0(4,2)          F2 <= J(1)*N(1)*CHT*CE = VD
         CDR    2,4               IF(VD.LT.VITH)
         BL     ARND              GOTO ARND
         LD     6,CONAN           F6 <= CONAN
         MDR    6,0               F6 <= CONAN*CHT
         MD     6,0(5,2)          F6 <= CONAN*N(1)*CHT
         LD     0,=D'1.'          F0 <= 1.D0
         DDR    4,2               F4 <= VITH/VD
         SDR    0,4               F0 <= 1.D0-VITH/VD
         MDR    6,0               F6 <= CONAN*N(1)*CHT*(1.D0-VITH/VD)
         AD     6,8(3,2)       F6 <= OSIG(1)+CONAN*N(1)*CHT*(1.D0-VITH/VD)
         STD    6,8(3,2)   OSIG(1)<=OSIG(1)+CONAN*N(1)*CHT*(1.D0-VITH/VD)
ARND     AR     2,8               R2 <= ADDR T(2)
LOOP     LH     12,0(2)           R12 <= HIGH ORDER 2 BYTES OF T(I)
         MVC    FLOAT+1(6),2(2) FLOAT <= FRACTIONAL DISPLACEMENT
         S      12,TDISP          REDUCE R12 BY TDISP. NOW # DOUBLE WORDS
*                                 FROM BASE OF INTERPOLATION TABLES
         LD     4,FLOAT           F4 <= FRACTION 0 LE FRAC LT 1
         BM     BADT              IF RESULT NEGATIVE - OUT OF RANGE
*                                 GOTO BADT
         C      12,TBND           IF R12 GREATER THAN TBND
         BH     BADT              GOTO BADT
         SD     4,=D'.5'          F4 <= FRAC -.5 LE FRAC LT .5
         SLA    12,3              R12 <= R2*8 NOW BYTE DISPLACEMENT
*                                 FROM BASE OF INTERPOLATION TABLES
*
*        NOW COMPUTE WEIGHTS FOR CUBIC INTERPOALTION OF
*        FUNCTIONS OF T
*
         LDR    2,4               F2 <= X
         MDR    4,4               F4 <= X**2 = X2
         HDR    4,4               F4 <= X2/2
         SD     4,=D'1.125'       F4 <= X2/2 - 9/8
         LDR    6,4               F6 <= X2/2 - 9/8
         HDR    4,4               F4 <= X2/4 - 9/16
         MDR    6,2               F6 <= X3/2 - 9X/8
         LCDR   0,4               F0 <= -X2/4 + 9/16
         ADR    0,6               F0 <= X3/2 - X2/4 - 9X/8 + 9/16
         STD    0,WM1             WM1 <= WEIGHT FOR TABLE ENTRY CORRESPOND-
*                                 ING TO CLOSEST SMALLER VALUE OF T.
         LCDR   0,4               F0 <= -X2/4 + 9/16
         SDR    0,6               F0 <= -X3/2 - X2/4 + 9X/8 + 9/16
         STD    0,WP1             WP1 <= WEIGHT FOR TABLE ENTRY CORRESPOND-
*                                 ING TO CLOSEST LARGER VALUE OF T.
         ADR    6,2               F6 <= X3/2 - X/8
         AD     4,=D'.5'          F4 <= X2/4 - 1/16
         MD     6,=X'4055555555555555'  F6 <= X3/6 - X/24
         LDR    0,4               F0 <= X2/4 - 1/16
         SDR    0,6               F0 <= -X3/6 + X2/4 + X/24 - 1/16
         ADR    6,4               F6 <= X3/6 + X2/4 - X/24 - 1/16
         STD    0,WM3             WM3 <= WEIGHT FOR SMALLEST VALUE OF T
         STD    6,WP3             WP3 <= WEIGHT FOR LARGEST VALUE OF T
*
* NOW CALCULATE INTERPOLATED VALUES OF TL AND TM
* (HAVE WEIGHTS FOR TABLES ENTRIES 1 & 4 IN FPR'S 0 & 6)
*
         LDR    4,0               F4 <= WEIGHT 1
         LDR    2,6               F2 <= WEIGHT 4
         MD     0,0(10,12)        F0 <= WEIGHT 1 * TL 1
         MD     2,24(10,12)       F2 <= WEIGHT 4 * TL 4
         MD     4,0(11,12)        F4 <= WEIGHT 1 * TM 1
         MD     6,24(11,12)       F6 <= WEIGHT 4 * TM 4
         ADR    0,2               F0 <= W1*TL1 + W4*TL4
         ADR    4,6               F4 <= W1*TM1 + W4*TM4
         LD     2,WM1             F2 <= WEIGHT 2
         LDR    6,2               F6 <= WEIGHT 2
         MD     2,8(10,12)        F2 <= W2*TL2
         MD     6,8(11,12)        F6 <= W2*TM2
         ADR    0,2               F0 <= W1*TL1 + W4*TL4 + W2*TL2
```

```
         ADR    4,6             F4 <= W1*TM1 + W4*TM4 + W2*TM2
         LD     2,WP1           F2 <= WEIGHT 3
         LDR    6,2             F6 <= WEIGHT 3
         MD     2,16(10,12)     F2 <= W3*TL3
         MD     6,16(11,12)     F6 <= W3*TM3
         ADR    0,2             F0 <= INTERPOLATED VALUE OF TL
         ADR    4,6             F4 <= INTERPOLATED VALUE OF TM
         MD     4,0(6,2)        F4 <= TM*LN(I)
         ADR    0,4             F0 <= TL + TM*LN(I)
         L      10,CNVA         R10 <= BASE ADDR CHINV TABLE
         STD    0,8(3,2)        OSIG(I) <= TL + TM*LN(I)
         L      11,VTHA         R11 <= BASE VTH TABLE
         AD     0,0(3,2)        F0 <= OSIG(I)+OSIG(I-1)
         MD     0,0(7,2)        F0 <= (OSIG(I)+OSIG(I-1))*SD(I)
         AD     0,TSI           F0 <= TSI + (OSIG(I)+OSIG(I-1))*SD(I)
         STD    0,TSI           TSI <= TSI + (OSIG(I)+OSIG(I-1))*SD(I)
         MD     0,TEMP3         F0 <= TSI*TEMP3
         AD     0,TEMP2         F0 <= TEMP2 + TEMP3*TSI
         STD    0,JCON          JCON <= TEMP2+TEMP3*TSI
         LD     6,=D'1.'        F6 <= 1.D0
         DDR    6,0             F6 <= 1.D0/(TEMP2+TEMP3*TSI)
         L      15,=V(F35)      R15 <= ENTRY ADDR F35
         STD    6,F35C1         F35C1 <= 1.D0/(TEMP2+TEMP3*TSI)
         STD    6,F35A          F35 ARGUMENT <= 1.D0/(TEMP2+TEMP3*TSI)
         BALR   14,15           F0 <= F35(1.D0/(TEMP2+TEMP3*TSI))
         STD    0,F35C2         F35C2 <= F35(1.D0/(TEMP2+TEMP3*TSI))
         LD     0,WM3           F0 <= WEIGHT FOR SMALLEST VALUE OF T
         LD     6,WP3           F6 <= WEIGHT FOR LARGEST VALUE OF T
*
* NOW CALCULATE INTERPOLATED VALUES OF CH AND VT
* (HAVE WEIGHTS FOR TABLES ENTRIES 1 & 4 IN FPR'S 0 & 6)
*
         LDR    4,0             F4 <= WEIGHT 1
         LDR    2,6             F2 <= WEIGHT 4
         MD     0,0(10,12)      F0 <= WEIGHT 1 * CH 1
         MD     2,24(10,12)     F2 <= WEIGHT 4 * CH 4
         MD     4,0(11,12)      F4 <= WEIGHT 1 * VT 1
         MD     6,24(11,12)     F6 <= WEIGHT 4 * VT 4
         ADR    0,2             F0 <= W1*CH1 + W4*CH4
         ADR    4,6             F4 <= W1*VT1 + W4*VT4
         LD     2,WM1           F2 <= WEIGHT 2
         LDR    6,2             F6 <= WEIGHT 2
         MD     2,8(10,12)      F2 <= W2*CH2
         MD     6,8(11,12)      F6 <= W2*VT2
         ADR    0,2             F0 <= W1*CH1 + W4*CH4 + W2*CH2
         ADR    4,6             F4 <= W1*VT1 + W4*VT4 + W2*VT2
         LD     2,WP1           F2 <= WEIGHT 3
         LDR    6,2             F6 <= WEIGHT 3
         MD     2,16(10,12)     F2 <= W3*CH3
         MD     6,16(11,12)     F6 <= W3*VT3
         ADR    0,2             F0 <= INTERPOLATED VALUE OF CH
         ADR    4,6             F4 <= INTERPOLATED VALUE OF VT
         MD     0,0(5,2)        F0 <= N(I)*CHT
         LDR    6,0             F6 <= N(I)*CHT = NCONO
         STD    0,NCONO         NCONO <= N(I)*CHT
         MD     6,0(7,2)        F6 <= NCONO*SD(I)
         MD     0,CE            F0 <= N(I)*CHT*CE
         MD     6,CONAN         F6 <= CONAN*NCONO*SD(I)
         STD    0,NCON          NCON <= N(I)*CHT*CE
         STD    6,CJA           CJA <= CONAN*NCONO*SD(I)
         LD     2,TEMP1         F2 <= TEMP1
         MD     2,F35C2         F2 <= TEMP1*F35C2 = J
         L      11,TMA          R11 <= BASE TM TABLE
         STD    2,0(4,2)        J(I) <= TEMP1*F35C2
         MDR    2,0             F2 <= JT*NCON = VD
         L      10,TLA          R10 <= BASE TL TABLE
         CDR    2,4             IF(VD.LT.VITH)
         BL     ARND1           GOTO ARND1
*
```

```
*         THE FOLLOWING CALCULATES THE CURRENT AND RESISTIVITY IN THE
*         CASE THE DRIFT VELOCITY EXCEEDS 13 TIMES THE ION THERMAL
*         VELOCITY - - IN THIS CASE AT LEAST PART OF THE SLAB
*         IS CHARACTERIZED BY ANOMALOUS RESISTIVITY
*
          LD     2,CJA          F2 <= CJA
          MD     2,TEMP3        F2 <= TEMP3*CJA
          LDR    6,4            F6 <= VITH
          STD    2,CJA          CJA <= TEMP3*CJA
          AD     2,JCON         F2 <= JCON + TEMP3*CJA
          STD    2,JCON         JCON <= JCON + TEMP3*CJA
          DDR    6,0            F6 <= (12./13.)*VITH/NCON = JA
          LD     2,CJA          F2 <= CJA
          MD     0,CONAN1       F0 <= NCON*CONAN1 = NCON1
          MDR    2,6            F2 <= CJA*JA
          STD    6,JA           JA <= (12./13.)*VITH/NCON
          STD    2,CJA          CJA <= CJA*JA
          LD     2,0(4,2)       F2 <= J(I)
          MD     0,0(7,2)       F0 <= NCON1*SD(I)
          MD     0,F35C1        F0 <= NCON1*SD(I)*F35C1
          MDR    0,2            F0 <= NCON1*SD(I)*F35C1*J(I)
          ADR    0,6            F0 <= NCON*SD(I)*F35C1*J(I)+JA
          SDR    2,6            F2 <= J(I)-JA
          DDR    2,0          F2 <= (J(I)-JA)/(NCON1*SD(I)*F35C1*J(I)+JA)
          AD     2,=D'1.'       F2 <= 1.D0 +     (J(I)-JA)
*                               /(NCON1*SD(I)*F35C1*J(I)+JA) = MULCON
          MDR    2,6            F2 <= JA*MULCON
          STD    2,0(4,2)       J(I) <= JA*MULCON
          L      11,COUNT       R11 <= COUNT(MAX # NEWTON'S METHOD STEPS)
NEWT      LD     6,CJA          F6 <= CJA
          DDR    6,2            F6 <= CJA/J(I) = CJAT
          L      15,=V(F35)     R15 <= ENTRY ADDRESS F35
          LD     0,JCON         F0 <= JCON
          SDR    0,6            F0 <= JCON - CJAT = JC1
          STD    0,F35A         F35A <= JC1
          MD     6,TCON         F6 <= TCON*CJAT = TC1
          STD    6,TC1          TC1 <= TCON*CJAT
          BALR   14,15          F0 <= F35(JC1)
          LD     4,TC1          F4 <= TC1
          LD     6,F35A         F6 <= JC1
          MDR    0,6            F0 <= F35(JC1)*JC1
          LD     2,0(4,2)       F2 <= J(I)
          MDR    0,2            F0 <= J(I)*JC1*F35(JC1)
          MD     6,TEMP1        F6 <= TEMP1*JC1
          ADR    6,4            F6 <= TC1 + TEMP1*JC1
          ADR    0,4            F0 <= TC1+J(I)*JC1*F35(JC1)
          DDR    6,0          F6 <= (TC1+TEMP1*JC1)/(TC1+J(I)*JC1*F35(JC1))
          MDR    2,6            F2 <= J(I)*(TC1+TEMP1*JC1)/
*                               (TC1+J(I)*JC1*F35(JC1))
          LDR    4,2            F4 <= NEW ESTIMATE OF J(I)
          LPDR   0,2            F0 <= DABS(NEW ESTIMATE OF J(I))
          SD     4,0(4,2)       F4 <= NEW ESTIMATE - OLD ESTIMATE
          MD     0,ERR          F0 <= ERR*DABS(NEW ESTIMATE)
          LPDR   4,4            F4 <= DABS(NEW ESTIMATE - OLD ESTIMATE)
          CDR    0,4            IF(DABS(NEW ESTIMATE - OLD ESTIMATE)
*                               /DABS(NEW ESTIMATE).LE.ERR)
          STD    2,0(4,2)       J(I) <= J(I)*(TC1+TEMP1*JC1)/
*                               (TC1+J(I)*JC1*F35(JC1))
          BNL    OUT            GOTO OUT
          BCT    11,NEWT
OUT       LD     4,JA           F4 <= JA
          DDR    4,2            F4 <= JA/J(I)
          L      11,TMA         R11 <= BASE ADDR TM ARRAY
          LD     2,=D'1.'       F2 <= 1.D0
          SDR    2,4            F2 <= 1.D0 - VITH/VD
          MD     2,NCONO        F2 <= NCONO*(1.D0-VITH/(J(I)*NCON))
          MD     2,CONAN        F2 <= CONAN*NCONO*(1.D0-VITH/(J(I)*NCON))
*                                = TSIG
          LDR    4,2            F4 <= TSIG
```

```
         AD     2,8(3,2)        F2 <= OSIG(I) + TSIG
         MD     4,0(7,2)        F4 <= TSIG*SD(I)
         AD     4,TSI           F4 <= TSI + TSIG*SD(I)
         STD    2,8(3,2)        OSIG(I) <=  OSIG(I) ÷ TSIG
         STD    4,TSI           TSI <= TSI + TSIG*SD(I)
ARND1    BXLE   2,8,LOOP        I <= I+1 AND GOTO LOOP IF NOT DONE
         L      13,4(13)        R13 <= ADDR OLD SAVE AREA
         LM     14,12,12(13)    GPR'S RESTORED
         SR     15,15           R15 <= 0 (RETURN CODE)
         MVI    12(13),X'FF'    INDICATE CONTROL RETURNED
         BR     14              RETURN
BADT     L      13,4(13)        R13 <= ADDR OLD SAVE AREA
         LM     14,12,12(13)    GPR'S RESTORED
         LA     15,4            R15 <= 4 (RETURN CODE)
         MVI    12(13),X'FF'    INDICATE CONTROL RETURNED
         BR     14              RETURN
         END
```

```
F35      CSECT
*
*        REAL FUNCTION F35*8(X)
*        F35 RETURNS THE 3./5. POWER OF THE ARGUMENT IF
*        THE ARGUMENT IS POSITIVE AND THE NEGATIVE OF THE 3./5.
*        POWER OF THE ABSOLUTE VALUE OF THE ARGUMENT IF THE ARGUMENT
*        IS NEGATIVE.  THE COMMENTS REFER TO THE POSITIVE CASE.
*        THE ALGORITHM IS:
*
*        CUBE X AND CALL THE RESULT Y, THEN WRITE Y AS
*
*        Y = (16**(5*N)) * (16**M) * (Z)
*
*        WHERE M IS BETWEEN -4 AND +4 AND Z IS BETWEEN 1/16 AND 1.
*        THEN Z**(1/5) IS APPROXIMATED BY A MINI-MAX LINEAR FIT
*        FROM TWO TABLES WITH A MAXIMUM RELATIVE ERROR IN THE
*        APPROXIMATION OF 5.1E-4.  THEN THE INITIAL ESTIMATE OF
*
*        T**1/5 = (16**(M/5)) * (Z**1/5)
*
*        IS REFINED BY TWO APPLICATIONS OF NEWTON'S METHOD.
*        X**3/5 IS THEN CALCULATED FROM (16**N) * (T**1/5).
*
         USING *,15             TELL ASSEMBLER NEXT INST ADDR IN R15
         B     FIRST            BRANCH AROUND NAME AND SAVE AREA
         DC    X'03'            LENGTH OF NAME
         DC    CL3'F35'         NAME
AREA     DS    18F              SAVE AREA
FIRST    STM   14,12,12(13)     SAVE CALLING ROUTINE'S GPR'S
         L     1,0(1)           R1 <= ADDR ARGUMENT (X)
         LD    4,0(1)           F4 <= X
         LR    9,13             R9 <= ADDR OLD SAVE AREA
         STD   4,ARG            ARG <= X
         LA    13,AREA          R13 <= ADDR NEW SAVE AREA
         DROP  15               R15 NO LONGER BASE REG
         USING AREA,13          R13 NEW BASE REG
         LTDR  4,4              CHECK SIGN OF X
         ST    9,4(13)          LINK SAVE AREAS
         BNM   NONNEG           IF SIGN POSITIVE - NO FIXES OR FLAGS
         NI    ARG,X'7F'        TURN OFF SIGN BIT OF ARG <= |X|
         O     9,=X'80000000'             SIGN BIT R9 ON - FLAG
NONNEG   ST    13,8(9)          LINK SAVE AREAS
         SR    3,3              R3 <= 0
         IC    3,ARG            R3 <= EXCESS 64 EXPONENET OF ARG
         MVI   ARG,X'40'        ARG <= FRACTION OF ARG
         LD    4,ARG            F4 <= FRACTION OF ARG
         L     4,=F'64'         R4 <= HEX 40
         SR    3,4              R3 <= EXPONENT OF ARG
         MDR   4,4              F4 <= FRACTION OF ARG **2
         M     2,=F'3'          R3 <= EXPONENT OF ARG**3
         LA    6,TAB1-16        R6 <= ADDR TABLE 1  -  16
         MD    4,ARG            F4 <= FRACTION OF ARG **3
         SR    5,5              R5 <= 0
         LA    7,TAB2-16        R7 <= ADDR TABLE 2  -  16
         STD   4,ARG            ARG <= FRACTION ARG **3
         IC    5,ARG            R5 <= EXCESS 64 EXPONENT OF
*                               FRACTION OF ARG **3
         AR    3,5              R3 <= EXCESS 64 EXPONENT OF |X|**3
         SR    2,2              R2 <= 0
         SR    3,4              R3 <= EXPONENT OF |X|**3
         LA    8,TAB3+16        R8 <= ADDR TABLE 3 + 16
         BNM   NOEXTD           IF R3 > 0 NO SIGN EXTEND
         L     2,=F'-1'         SIGN EXTEND FOR DIVIDE
NOEXTD   D     2,=F'5'          R2 <= EXPONENT OF T
*                               R3 <= N (SEE COMMENTS)
```

```
         A      3,=F'65'        R3 <= EXCESS 64 EXPONENT OF 16**N
         AR     4,2             R4 <= EXCESS 64 EXPONENT OF T
         STC    4,ARG           ARG <= T
         LD     4,ARG           F4 <= T
         SLL    2,2             R2 <= DISPLACEMENT FOR TABLE 3
         SDR    0,0             F0 <= 0.D0
         L      5,ARG           R5 <= 1ST 4 BYTES OF T
         IC     4,ARG+1         LOW ORDER BYTE R4 <= FIRST BYTE OF
*                               FRACTION OF T
         SLL    5,8             HIGH ORDER 3 BYTES OF R5 <= BYTES
*                               1-3 OF FRACTION OF T
         N      4,=X'000000FC'            R4 <= DISPLACE FOR TABLES 1 & 2
         IC     5,ARG+4         R5 <= BYTES 1-4 OF FRACTION OF T
         LE     0,0(4,7)        F0 <= TABLE 2 ENTRY - SLOPE
         SRL    5,2             LOW 3 BYTES OF R3 <= FRACTION OF
*        FRACTIONAL DISPLACEMENT FOR MINI-MAX LINE
         ST     5,FRAC          FRAC <= FRACTIONAL DISPLACEMENT FRACTION
         MVI    FRAC,X'40'      FRAC <= FRACTIONAL DISPLACEMENT
         ME     0,FRAC          F0 <= FRAC*SLOPE
         AE     0,0(4,6)        F0 <= MINI-MAX ESTIMATE OF Z**1/5
         LTR    9,9             CHECK IF X NEGATIVE
         ME     0,0(2,8)        F0 <= MINI-MAX EST OF T**1/5
         BNM    NOSIGN          IF X POSITIVE, NO FIXES
         O      3,=F'128'       SIGN OF EXPONENT OF 16**N MADE MINUS
NOSIGN   LDR    2,0             F2 <= MINI-MAX EST OF T**1/5 = EST 1
*
*        BEGIN TWO APPLICATIONS OF NEWTON'S METHOD
*        WITH SOME GPR FIX-UPS INTERLEAVED.
*
         MDR    2,2             F2 <= EST1 ** 2
         STC    3,MUL           MUL <= SIGN(X) * 16**N
         MDR    2,2             F2 <= EST 1 ** 4
         L      1,24(9)         R1 RESTORED
         LDR    6,2             F6 <= EST 1 ** 4
         MDR    2,0             F2 <= EST 1 ** 5
         L      2,28(9)         R2 RESTORED
         SDR    2,4             F2 <= EST 1 ** 5 - T
         DER    2,6             F2 <= (EST 1 ** 5 - T)/EST 1 ** 4
         L      3,32(9)         R3 RESTORED
         ME     2,ONE5          F2 <= (EST 1 ** 5 - T)/5*EST 1 ** 4
         L      4,36(9)         R4 RESTORED
         SDR    0,2             F0 <= EST 2
         LDR    2,0             F2 <= EST 2
         MDR    2,2             F2 <= EST 2 ** 2
         L      5,40(9)         R5 RESTORED
         MDR    2,2             F2 <= EST 2 ** 4
         L      6,44(9)         R6 RESTORED
         LDR    6,2             F6 <= EST 2 ** 4
         MDR    2,0             F2 <= EST 2 ** 5
         L      7,48(9)         R7 RESTORED
         SDR    2,4             F2 <= EST 2 ** 5 - T
         L      8,52(9)         R8 RESTORED
         DDR    2,6             F2 <= (EST 2 ** 5 - T)/EST 2 ** 4
         SR     15,15           R15 <= 0 (RETURN CODE)
         MD     2,ONE5          F2 <= (EST 2 ** 5 - T)/5*EST 2 ** 4
         MVI    12(9),X'FF'     INDICATE CONTROL RETURNED
         SDR    0,2             F0 <= EST 3 (FINAL ESTIMATE OF T**1/5)
         L      9,56(9)         R9 RESTORED
         MD     0,MUL           F0 <= ESTIMATE OF |X|**3/5
         L      13,4(13)        R13 RESTORED
         BR     14              RETURN
         CNOP   4,8             FORCE CORRECT ALIGNMENT
FRAC     DC     F'0'
ARG      DC     D'0.'
MUL      DC     X'0010000000000000'
ONE5     DC     X'4033333333333333'
         LTORG
TAB1     DC     X'40931BB3'
         DC     X'4099CBC3'
```

```
         DC    X'409F7DF5'
         DC    X'40A479CE'
         DC    X'40A8EBB4'
         DC    X'40ACF139'
         DC    X'40B09F20'
         DC    X'40B40494'
         DC    X'40B72D05'
         DC    X'40BA214F'
         DC    X'40BCE873'
         DC    X'40BF8816'
         DC    X'40C204D6'
         DC    X'40C46286'
         DC    X'40C6A45E'
         DC    X'40C8CD17'
         DC    X'40CADF05'
         DC    X'40CCDC2C'
         DC    X'40CEC64A'
         DC    X'40D09EE7'
         DC    X'40D2675C'
         DC    X'40D420DA'
         DC    X'40D5CC70'
         DC    X'40D76B12'
         DC    X'40D8FD98'
         DC    X'40DA84C8'
         DC    X'40DC0153'
         DC    X'40DD73DA'
         DC    X'40DEDCF1'
         DC    X'40E03D1D'
         DC    X'40E194DA'
         DC    X'40E2E499'
         DC    X'40E42CC1'
         DC    X'40E56DB3'
         DC    X'40E6A7C8'
         DC    X'40E7DB51'
         DC    X'40E9089C'
         DC    X'40EA2FF0'
         DC    X'40EB518F'
         DC    X'40EC6DB7'
         DC    X'40ED84A1'
         DC    X'40EE9686'
         DC    X'40EFA397'
         DC    X'40F0AC04'
         DC    X'40F1AFFB'
         DC    X'40F2AFA6'
         DC    X'40F3AB2D'
         DC    X'40F4A2B6'
         DC    X'40F59664'
         DC    X'40F6865B'
         DC    X'40F772B9'
         DC    X'40F85B9D'
         DC    X'40F94124'
         DC    X'40FA236A'
         DC    X'40FB0288'
         DC    X'40FBDE98'
         DC    X'40FCB7B1'
         DC    X'40FD8DEA'
         DC    X'40FE6159'
         DC    X'40FF3211'
TAB2     DC    X'3F6B5E5D'
         DC    X'3F5B5AA7'
         DC    X'3F4FE171'
         DC    X'3F4736EF'
         DC    X'3F4069F9'
         DC    X'3F3AEB91'
         DC    X'3F366146'
         DC    X'3F328EEF'
         DC    X'3F2F4AE5'
         DC    X'3F2C7758'
         DC    X'3F29FE5F'
```

```
         DC      X'3F27CF79'
         DC      X'3F25DDFD'
         DC      X'3F242004'
         DC      X'3F228DB5'
         DC      X'3F2120C2'
         DC      X'3F1FD408'
         DC      X'3F1EA350'
         DC      X'3F1D8B17'
         DC      X'3F1C8870'
         DC      X'3F1B98DF'
         DC      X'3F1ABA4B'
         DC      X'3F19EAE6'
         DC      X'3F192921'
         DC      X'3F1873A6'
         DC      X'3F17C948'
         DC      X'3F172902'
         DC      X'3F1691EC'
         DC      X'3F16033B'
         DC      X'3F157C3C'
         DC      X'3F14FC4D'
         DC      X'3F1482E0'
         DC      X'3F140F75'
         DC      X'3F13A198'
         DC      X'3F1338E1'
         DC      X'3F12D4F4'
         DC      X'3F12757A'
         DC      X'3F121A27'
         DC      X'3F11C2B5'
         DC      X'3F116EE4'
         DC      X'3F111E78'
         DC      X'3F10D13D'
         DC      X'3F108701'
         DC      X'3F103F97'
         DC      X'3EFFAD4E'
         DC      X'3EFB8941'
         DC      X'3EF78B10'
         DC      X'3EF3B0AD'
         DC      X'3EEFF82D'
         DC      X'3EEC5FC9'
         DC      X'3EE8E5DA'
         DC      X'3EE588D6'
         DC      X'3EE2474D'
         DC      X'3EDF1FE8'
         DC      X'3EDC1167'
         DC      X'3ED91A9C'
         DC      X'3ED63A6E'
         DC      X'3ED36FD6'
         DC      X'3ED0B9DC'
         DC      X'3ECE1796'
TAB3     DC      X'401BDB8C'
         DC      X'403080C0'
         DC      X'405472D1'
         DC      X'4093088C'
         DC      X'41100000'
         DC      X'411BDB8C'
         DC      X'413080C0'
         DC      X'415472D1'
         DC      X'4193088C'
         END
```

```
TESTP    CSECT
*
*        ROUGHLY EQUIVALENT TO THE FORTRAN CODE BELOW EXCEPT THE
*        THE FUNCTION CHINV PASSED IN THE ARGUMENT LIST
*        IS IMPLEMENTED IN LINE IN THE ASSEMBLY LANGUAGE VERSION.
*        A CALL TO THE ASSEMBLY LANGUAGE VERSION SHOULD
*        PASS A SEMI-LOGARITHMIC INTERPOLATION TABLE (CHINV(884))
*        RATHER THAN A FUNCTION NAME.  ALSO THE PARAMETER ADDRESSES
*        ARE OBTAINED FROM LOCAL STORAGE IN THE CALL TO TESTP
*        NOT FROM THE PARAMETER LIST.  THE PARAMETER ADDRESSES
*        ARE INITIALIZED BY THE ENTRY POINT TESTPI IN THE ASSEMBLY
*        LANGUAGE VERSION.
*
*        NOTE THAT THIS MEANS TESTPI MUST BE CALLED BEFORE
*        THE FIRST CALL TO TESTP OR UNPREDICTABLE ABENDS
*        WILL RESULT.
*
*     TESTP(T,TE,J,OSIG,N,TUP,DELT,FRAC,NTAB,CHINV)
*     IMPLICIT REAL*8 (A-H,O-Z)
*     REAL*8 T(NTAB),TE(NTAB),J(NTAB),OSIG(NTAB),N(NTAB),TUP(NTAB)
*     REAL*8 C/2.997925D10/FRAC,FRACI,DELT,DELTS
*     GOTO 10
*     ENTRY TESTPI(T,TE,J,OSIG,N,TUP,DELT,FRAC,NTAB,CHINV)
*     FRACI=2.D0/FRAC
*     RETURN
*  10 DELTS=DELT*FRACI
*     DO 20 I=1,NTAB
*     TUP(I)=C*J(I)*J(I)*OSIG(I)*N(I)
*     IF(DELTS*TUP(I).LE.TE(I))GOTO 20
*     DELTS=TE(I)/TUP(I)
*  20 CONTINUE
*     DELT=FRAC*DELTS
*     DO 20 I=1,NTAB
*     TE(I)=TE(I)+TUP(I)*DELT
*  20 T(I)=CHINV(TE(I))
*     RETURN
*     END
*
         USING *,15
         B     TFIRST         BRANCH AROUND NAME, SAVE AREA ETC.
         DC    X'05'
         DC    CL5'TESTP'
         ENTRY TESTPI
         USING *,15
TESTPI   B     IFIRST         BRANCH AROUND NAME,SAVE AREA ETC.
         DC    X'06'
         DC    CL7'TESTPI '
AREA     DS    18F            SAVE AREA
TDISP    DC    X'00004410'
TBND     DC    X'00000230'
REGS     DS    10F            REGISTER STORAGE
         CNOP  0,8            FORCE DOUBLE WORD  ALIGNMENT
C        DC    D'2.997925E10'
TION     DC    X'4519AEF8FF9F9C62'
FRAC     DC    D'0.'
FRACI    DC    D'0.'
DELT     DC    D'0.'
WM1      DC    D'0.'
WP1      DC    D'0.'
FLOAT    DC    X'4000000000000000'
*
*        FIRST ENTRY POINT
*
IFIRST   STM   14,12,12(13)   SAVE CALLING ROUTINE'S GPR'S
         LR    2,13           R2 <= ADDR OLD SAVE AREA
```

```
         LA    13,AREA         R13 <= ADDR NEW SAVE AREA
         DROP  15              R15 NO LONGER BASE REG
         USING AREA,13         R13 NEW BASE REGISTER
         ST    2,4(13)         LINK SAVE AREAS
         ST    13,8(2)
         LM    3,12,0(1)       R3-R12 <= ADDR'S ARGS
         LD    0,0(10)         F0 <= FRAC
         L     11,0(11)        R11 <= NTAB
         LD    2,=D'2.'        F2 <= 2.D0
         SLA   11,3            R11 <= NTAB*8
         STD   0,FRAC          FRAC(LOCAL) <= FRAC
         SR    3,4             R3 <= BASE T - BASE TE
         DDR   2,0             F2 <= 2.D0/FRAC
         SR    5,4             R5 <= BASE J - BASE TE
         STD   2,FRACI         FRACI <= 2.D0/FRAC
         SR    6,4             R6 <= BASE OSIG - BASE TE
         SR    7,4             R7 <= BASE N - BASE TE
         SR    8,4             R8 <= BASE TUP - BASE TE
         LA    10,8            R10 <= 8 (INCREMENT)
         SR    11,10           R11 <= 8*(NTAB-1)
         AR    11,4            R11 <= BASE TE + 8*(NTAB-1) COMPARAND
         STM   3,12,REGS       REGS <= R3-R12
         L     13,4(13)        R13 <= ADDR OLD SAVE AREA
         LM    14,12,12(13)    GPR'S RESTORED
         SR    15,15           R15 <= 0 (RETURN CODE)
         MVI   12(13),X'FF'    INDICATE CONTROL RETURNED
         BR    14              RETURN
         DROP  13
         USING TESTP,15
TFIRST   STM   14,12,12(13)    SAVE CALLING ROUTINE'S GPR'S
         LR    2,13            R2 <= ADDR OLD SAVE AREA
         LA    13,AREA         R13 <= ADDR NEW SAVE AREA
         DROP  15              R15 NO LONGER BASE REG
         USING AREA,13         R13 NEW BASE REGISTER
         ST    2,4(13)         LINK SAVE AREAS
         ST    13,8(2)
         LM    3,12,REGS       SET UP GPR'S
         LD    6,0(9)          F6 <= DELT
         MD    6,FRACI         F6 <= DELT*FRACI = DELTS
LOOP1    LD    0,0(5,4)        F0 <= J(I)
         MDR   0,0             F0 <= J(I)*J(I)
         MD    0,C             F0 <= C*J(I)*J(I)
         MD    0,0(6,4)        F0 <= C*J(I)*J(I)*OSIG(I)
         MD    0,0(7,4)        F0 <= C*J(I)*J(I)*OSIG(I)*N(I)
         STD   0,0(8,4)        TUP(I) <= C*J(I)*J(I)*OSIG(I)*N(I)
         MDR   0,6             F0 <= TUP(I)*DELTS
         CD    0,0(4)          IF(DELTS*TUP(I).LE.TE(I))
         BNH   ARND            GOTO ARND
         LD    6,0(4)          F6 <= TE(I)
         DD    6,0(8,4)        F6 <= TE(I)/TUP(I) = DELTS
ARND     BXLE  4,10,LOOP1      I <= I+1 & GOTO LOOP1 IF NOT DONE
         MD    6,FRAC          F6 <= DELTS*FRAC = DELT
         L     4,REGS+4        R4 <= BASE TE
         STD   6,0(9)          DELT <= DELTS*FRAC
LOOP2    LD    0,0(8,4)        F0 <= TUP(I)
         MD    0,0(9)          F0 <= TUP(I)*DELT
         AD    0,0(4)          F0 <= TE(I) + TUP(I)*DELT
         STD   0,0(4)          TE(I) <= TE(I) + TUP(I)*DELT
         LH    2,0(4)          R2 <= HIGH ORDER BYTES OF TE(I)
         MVC   FLOAT+1(6),2(4)         FLOAT <= FRACTIONAL DISPLACEMENT
         S     2,TDISP         REDUCE R2 BY TDISP NOW # DOUBLE WORDS
*                              FROM BASE OF INTERPOLATION TABLES
         LD    4,FLOAT         F4 <= FRACTION 0 LE FRAC LT 1
         BM    LOWT            IF RESULT NEGATIVE - OUT OF RANGE
*                              GOTO LOWT
         C     2,TBND          COMPARE R2 TO TBND IF GREATER
         BH    HITE            OUT OF RANGE - GOTO HITE
         SD    4,=D'.5'        F4 <= X = FRAC - .5  -.5 LE X LE .5
         SLA   2,3             R2 <= R2*8 NOW BYTE DISPLACEMENT
```

```
*                                 FROM BASE OF INTERPOLATION TABLE.
*
*        NOW COMPUTE WEIGHTS FOR CUBIC INTERPOALTION OF
*        FUNCTIONS OF T
*
         LDR   2,4             F2 <= X
         MDR   4,4             F4 <= X**2 = X2
         HDR   4,4             F4 <= X2/2
         SD    4,=D'1.125'     F4 <= X2/2 - 9/8
         LDR   6,4             F6 <= X2/2 - 9/8
         HDR   4,4             F4 <= X2/4 - 9/16
         MDR   6,2             F6 <= X3/2 - 9X/8
         LCDR  0,4             F0 <= -X2/4 + 9/16
         ADR   0,6             F0 <= X3/2 - X2/4 - 9X/8 + 9/16
         STD   0,WM1           WM1 <= WEIGHT FOR TABLE ENTRY CORRESPOND-
*                              ING TO CLOSEST SMALLER VALUE OF T.
         LCDR  0,4             F0 <= -X2/4 + 9/16
         SDR   0,6             F0 <= -X3/2 - X2/4 + 9X/8 + 9/16
         STD   0,WP1           WP1 <= WEIGHT FOR TABLE ENTRY CORRESPOND-
*                              ING TO CLOSEST LARGER VALUE OF T.
         ADR   6,2             F6 <= X3/2 - X/8
         AD    4,=D'.5'        F4 <= X2/4 - 1/16
         MD    6,=X'4055555555555555'  F6 <= X3/6 - X/24
         LDR   0,4             F0 <= X2/4 - 1/16
         SDR   0,6             F0 <= -X3/6 + X2/4 + X/24 - 1/16
         ADR   6,4             F6 <= X3/6 + X2/4 - X/24 - 1/16
         MD    0,0(12,2)       F0 <= WEIGHT1 * CHNV1
         MD    6,24(12,2)      F6 <= WEIGHT4 * CHNV4
         LD    2,WM1           F2 <= WWIGHT2
         ADR   0,6             F0 <= W1*CHNV1+W4*CHNV4
         LD    4,WP1           F4 <= WEIGHT3
         MD    2,8(12,2)       F2 <= W2*CHNV2
         MD    4,16(12,2)      F4 <= W3*CHNV3
         ADR   0,2             F0 <= W1*CHNV1+W2*CHNV2+W4*CHNV4
         ADR   0,4             F0 <= INTERPOLATED VALUE OF CHINV(TE(I))
         STD   0,0(3,4)        T(I) <= CHINV(TE(I))
         BXLE  4,10,LOOP2      I<=I+1 & GOTO LOOP2 IF NOT DONE
         B     ARND1
LOWT     STD   0,0(3,4)        T(I) <= TE(I) (FOR T < 4096 K)
         BXLE  4,10,LOOP2      I <= I+1 AND GOTO LOOP2 IF NOT DONE
         B     ARND1           GOTO ARND1
HITE     SD    0,TION          F0 <= TE(I) - TION
         HDR   0,0             F0 <= (TE(I)-TION)/2.D0
         STD   0,0(3,4)        T(I) <=  (TE(I)-TION)/2.D0
         BXLE  4,10,LOOP2      I <= I+1 AND GOTO LOOP2 IF NOT DONE
ARND1    L     13,4(13)        R13 <= ADDR OLD SAVE AREA
         LM    14,12,12(13)    GPR'S RESTORED
         SR    15,15           R15 <= 0 (RETURN CODE)
         MVI   12(13),X'FF'    INDICATE CONTROL RETURNED
         BR    14              RETURN
         END
```

ORIGINAL PAGE IS
OF POOR QUALITY

TOUT

```
TOUT     CSECT
*
*        ROUGHLY EQUIVALENT TO THE THREE FORTRAN SUBROUTINES BELOW.
*        NOUT OPENS LOGICAL UNIT 9 (FT09F001) AND DOES THE OUTPUT
*        OF THE SUBROUTINE NOUT.  THE PARAMETER NTAB IS PASSED
*        BY ENTRY POINT NOUT AND THE NTAB IN THE CALLING SEQUENCE
*        TO TOUT IS IGNORED BY THE ASSEMBLY LANGUAGE VERSION OF
*        THESE ROUTINES.  FORTRAN CLOSES DATA SETS IT KNOWS ABOUT
*        BUT FORTRAN WON'T KNOW ABOUT THIS DATA SET SO CTOUT MUST
*        BE CALLED BEFORE THE STOP STATEMENT IN THE MAIN ROUTINE.
*        THE FORTRAN CTOUT DOES THE SAME THING - I.E. IT CAUSES A
*        CLOSE TO BE ISSUED FOR THE DATA SET REFERENCED BY THE
*        DDNAME FT09F001.  USES QSAM UNDER OS/VS2.
*
*
*****************************************************************
********************** NOTE *************************************
*****************************************************************
****                                                         ****
****       NOUT MUST BE CALLED BEFORE TOUT IF TOUT IS TO     ****
****       WORK SINCE NOUT SETS UP AN AREA WITH THE          ****
****       REGISTERS THAT TOUT USES - THIS MAKES SENSE       ****
****       IN THIS APPLICATION (NOUT IS ALWAYS CALLED        ****
****       FIRST) BUT MUST BE CHANGED IF NOUT IS NOT TO BE   ****
****       CALLED BEFORE TOUT                                ****
****                                                         ****
*****************************************************************
*****************************************************************
*
*
*
*     SUBROUTINE NOUT(EFLUX,PSIO,FRAC,TIMMAX,NTAB,N,S)
*     IMPLICIT REAL*8 (A-H,O-Z)
*     REAL*8 N(NTAB),S(NTAB)
*9001 FORMAT(10A8)
*     WRITE(9,9001)EFLUX,PSIO,FRAC,TIMMAX,NTAB
*     WRITE(9,9001)N
*     WRITE(9,9001)S
*     RETURN
*     END
*
*     SUBROUTINE TOUT(TIM,T,J,DELT,IITER,NTAB)
*     IMPLICIT REAL*8 (A-H,O-Z)
*     REAL*8 T(NTAB),J(NTAB)
*9001 FORMAT(10A8)
*     WRITE(9,9001)TIM,DELT,IITER
*     WRITE(9,9001)T
*     WRITE(9,9001)J
*     RETURN
*     END
*
*     SUBROUTINE CTOUT
*     END FILE 9
*     RETURN
*     END
*
         USING *,15
         B     WFIRST           BRANCH AROUND NAME, VARIABLES, SAVE AREA
*                               AND OTHER ENTRY POINTS
         DC    X'04'            LENGTH OF NAME
         DC    CL5'TOUT '       NAME
         ENTRY NOUT
         USING *,15
NOUT     B     OFIRST           BRANCH AROUND NAME, VARIABLES, SAVE AREA
*                               AND OTHER ENTRY POINT
```

```
         DC    X'04'          LENGTH OF NAME
         DC    CL5'NOUT '     NAME
         USING *,15
         ENTRY CTOUT
CTOUT    B     CFIRST         BRANCH AROUND NAME, VARIABLES AND AREA
         DC    X'05'          LENGTH OF NAME
         DC    CL5'CTOUT'
         DROP  15
         USING AREA,13
AREA     DS    18F            SAVE AREA
BUF      DS    20F            OUTPUT BUFFER (ONE CARD - WE USE QSAM)
BLNCRD   DC    20CL4'    '    A BLANK CARD - I'M LAZY
TEN      DC    F'10'
LMOV3    MVC   BUF(1),0(3)    TO BE EXECUTED BY AN EX
LMOV4    MVC   BUF(1),0(4)
SREG     DS    5F
         DROP  13
         USING CTOUT,15
*
*        CTOUT
*
CFIRST   STM   14,12,12(13)   SAVE CALLING ROUTINE'S GPR'S
         LR    2,13           R2 <= ADDR OLD SAVE AREA
         LA    13,AREA        R13 <= ADDR NEW SAVE AREA
         DROP  15             R15 NO LONGER BASE REG
         USING AREA,13        R13 NEW BASE REG
         ST    2,4(13)        LINK SAVE AREAS
         ST    13,8(2)
         CLOSE LU9DCB
*
*        RETURN SEQUENCE
*
         L     13,4(13)       R13 <= ADDR OLD SAVE AREA
         LM    14,2,12(13)    GPR'S RESTORED
         SR    15,15          R15 <=0, RETURN CODE
         MVI   12(13),X'FF'   INDICATE CONTROL RETURNED
         BR    14             RETURN
         DROP  13
*
*        NOUT(EFLUX,PSIO,FRAC,TIMMAX,NTAB,N,S)
*
         USING NOUT,15
OFIRST   STM   14,12,12(13)   SAVE CALLING ROUTINE'S GPR'S
         LR    2,13           R2 <= ADDR OLD SAVE AREA
         LA    13,AREA        R13 <= ADDR NEW SAVE AREA
         DROP  15             R15 NO LONGER BASE REG
         USING AREA,13        R13 NEW BASE REG
         ST    2,4(13)        LINK SAVE AREAS
         ST    13,8(2)
         LM    2,8,0(1)       R2-R8 <= ADDR'S ARGS
         OPEN  (LU9DCB,(OUTPUT))
         MVC   BUF(80),BLNCRD           FILL BUFFER WITH BLANKS
         MVC   BUF(8),0(2)    FIRST 8 BYTES OF BUFFER <= EFLUX
         MVC   BUF+8(8),0(3)  2ND 8 BYTES OF BUFFER <= PISO
         MVC   BUF+16(8),0(4)           3RD 8 BYTES OF BUFFER <= FRAC
         MVC   BUF+24(8),0(5)           4TH I BYTES OF BUFFER <= TIMMAX
         MVC   BUF+36(4),0(6)           2ND HALF OF 5TH 8 BYTES OF
*                             BUFFER <= NTAB
         PUT   LU9DCB,BUF     WRITE OUT 1ST RECORD
         LR    3,7            R3 <= BASE ADDR N
         LR    4,8            R4 <= BASE ADDR S
         L     7,0(6)         R7 <= NTAB
         SR    6,6            R6 <= 0
         D     6,TEN          R6 <= REMAINDER OF NTAB/10
*                             R7 <= INTEGER PART OF NTAB/10
         LR    5,6            R5 <= REMAINDER OF NTAB/10
         M     6,TEN          R7 <= (NTAB/10)*10 INTEGER MODE
*                             # CARDS - 1 (UNLESS NTAB ENDS IN 0)
         SLL   5,3            F5 <= REMAINDER NTAB/10 * 8
```

```
         SLL   7,3              R7 <= (NTAB/10)*10 * 8
*                               # BYTES IN # CARDS - 1
         LA    6,80             R6 <= 80 (INCREMENT FOR LOOP)
         LR    2,3              R2 <= BASE ADDR N
         LA    4,0(4)           HIGH BYTE OF BASE ADDR S ZEROED
         SR    7,6              R7 <= (NTAB/10 - 1) * 80
         STM   5,7,SREG         SAVE INCREMENTS AND OFFSET FOR COMPARAND
         STM   6,7,SREG+12      TWO COPIES OF INCREMENTS AND OFFSET
         AR    7,3              R7 <= BASE ADDR N + ((NTAB/10)-1)*80
LOOP1    MVC   BUF(80),0(3)     PUT NEXT 80 BYTES IN BUFFER
*                               AND WRITE THEM OUT
         PUT   LU9DCB,BUF
         BXLE  3,6,LOOP1        KEEP GOING TILL WE'RE DONE
         LTR   5,5              CHECK IF NO MORE THINGS TO WRITE
         BZ    ARND1            IF NOT DON'T WRITE OUT ANOTHER CARD
*                               BECAUSE FORTRAN WOULDN'T
         MVC   BUF(80),BLNCRD            FILL BUFFER WITH BLANKS
         EX    5,LMOV3          MOVE IN LAST FEW (<10) VALUES
*                               AND WRITE OUT LAST CARD FOR N
         PUT   LU9DCB,BUF
ARND1    SR    7,2              R7 <= (NTAB/10 - 1) * 80
         AR    7,4              R7 <= BASE S + ((NTAB/10)-1)*80
LOOP2    MVC   BUF(80),0(4)     PUT NEXT 80 BYTES IN BUFFER
*                               AND WRITE THEM OUT
         PUT   LU9DCB,BUF
         BXLE  4,6,LOOP2        KEEP GOING TILL WE'RE DONE
         LTR   5,5              CHECK IF NO MORE THINGS TO WRITE
         BZ    ARND2            IF NOT DON'T WRITE OUT ANOTHER CARD
*                               BECAUSE FORTRAN WOULDN'T
         MVC   BUF(80),BLNCRD            FILL BUFFER WITH BLANKS
         EX    5,LMOV4          MOVE IN LAST FEW (<10) VALUES
*                               AND WRITE OUT LAST CARD FOR S
         PUT   LU9DCB,BUF
*
*        RETURN SEQUENCE
*
ARND2    L     13,4(13)         R13 <= ADDR OLD SAVE AREA
         LM    14,8,12(13)      GPR'S RESTORED
         SR    15,15            R15 <= 0, RETURN CODE
         MVI   12(13),X'FF'     INDICATE CONTROL RETURNED
         BR    14               RETURN
         DROP  13
*
*        TOUT(TIM,T,J,DELT,IITER,NTAB)  WE IGNORE LAST PARAMETER
*
         USING TOUT,15
WFIRST   STM   14,12,12(13)     SAVE CALLING ROUTINE'S GPR'S
         LR    2,13             R2 <= ADDR OLD SAVE AREA
         LA    13,AREA          R13 <= ADDR NEW SAVE AREA
         DROP  15               R15 NO LONGER BASE REG
         USING AREA,13          R13 NEW BASE REG
         ST    2,4(13)          LINK SAVE AREAS
         ST    13,8(2)
         LM    2.6,0(1)          R2-R6 <= ADDR'S ARG'S WE USE
         MVC   BUF(80),BLNCRD            FILL BUFFER WITH BLANKS
         MVC   BUF(8),0(2)      FIRST 8 BYTES OF BUFFER <= TIM
         MVC   BUF+8(8),0(5) 2ND 8 BYTES <= DELT
         MVC   BUF+20(4),0(6) 2ND HALF 3 8 BYTES <= IITER
         PUT   LU9DCB,BUF
         LM    5,9,SREG         R5-R9 <= NUMBER OF VALUES ON LAST CARD FOR
*                               T AND J AND INCREMENTS AND COMPARANDS
         AR    7,3              R7 <= COMPARAND FOR LOOP3
         AR    9,4              R9 <= COMPARAND FOR LOOP4
LOOP3    MVC   BUF(80),0(3)     PUT NEXT 80 BYTES IN BUFFER
*                               AND WRITE THEM OUT
         PUT   LU9DCB,BUF
         BXLE  3,6,LOOP3        KEEP GOING TILL WE'RE DONE
         LTR   5,5              CHECK IF NO MORE THINGS TO WRITE
         BZ    LOOP4            IF NOT DON'T WRITE OUT ANOTHER CARD
```

```
*                                  BECAUSE FORTRAN WOULDN'T
         MVC    BUF(80),BLNCRD              FILL BUFFER WITH BLANKS
         EX     5,LMOV3            MOVE IN LAST FEW (<10) VALUES
*                                  AND WRITE OUT LAST CARD FOR T
         PUT    LU9DCB,BUF
LOOP4    MVC    BUF(80),0(4)       PUT NEXT 80 BYTES IN BUFFER
*                                  AND WRITE THEM OUT
         PUT    LU9DCB,BUF
         BXLE   4,8,LOOP4          KEEP GOING TILL WE'RE DONE
         MVC    BUF(80),BLNCRD              FILL BUFFER WITH BLANKS
         LTR    5,5                CHECK IF NO MORE THINGS TO WRITE
         BZ     ARND3              IF NOT DON'T WRITE OUT ANOTHER CARD
*                                  BECAUSE FORTRAN WOULDN'T
         EX     5,LMOV4            MOVE IN LAST FEW (<10) VALUES
*                                  AND WRITE OUT LAST CARD FOR J
         PUT    LU9DCB,BUF
*
*        RETURN SEQUENCE
*
ARND3    L      13,4(13)           R13 <= ADDR OLD SAVE AREA
         LM     14,9,12(13)        GPR'S RESTORED
         SR     15,15              R15 <= 0, RETURN CODE
         MVI    12(13),X'FF'       INDICATE CONTROL RETURNED
         BR     14                 RETURN
LU9DCB DCB DEVD=DA,MACRF=PM,DSORG=PS,RECFM=FB,LRECL=80,DDNAME=FT09F001
         END
```

## Appendix B

## STEADY STATE MODEL OF THE SOLAR ATMOSPHERE

We have constructed a steady state numerical model of the solar atmosphere. The model was developed to investigate the effects of upward velocities and diverging magnetic field patterns on the temperature and density structure of the solar atmosphere; however, for this work the model is used only to provide reasonable temperature and density profiles for the estimation of the effect of reverse current heating on the atmosphere. The computer program calculates the run of temperature and density in an individual flux tube.

The equations governing the behavior of an inviscid compressible fluid in the presence of gravity are

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \quad , \tag{B.1}$$

$$\frac{\partial}{\partial t} (\rho \vec{u}) + \nabla \cdot (\rho \vec{u}\vec{u}) = - \nabla P + \rho g \quad , \tag{B.2}$$

$$\frac{\partial}{\partial t} (\rho \epsilon) + \nabla \cdot (\rho \epsilon \vec{u}) = - \nabla \cdot \vec{q} - P\nabla \cdot \vec{u} - \mathcal{L} + S \quad , \tag{B.3}$$

where $\epsilon$ is the total internal energy per unit mass, $\vec{q}$ is the heat flux, $g$ is the gravitational acceleration, $\mathcal{L}$ is the energy lost via radiation, and $S$ is the sum of all other non-thermal energy sources or sinks. For flow along a magnetic flux tube, considering variation only along the field lines and assuming the radius of curvature of the field lines to be large compared to the dimensions of the flux tube, we see that the equations become one-dimensional. If we add the definition of the heat flux and an equation of state to Equations (B.1)-(B.3), we may write a complete set of equations for the steady state ($\partial/\partial t = 0$) case

$$u_s \frac{d\rho}{ds} + \rho \frac{du_s}{ds} + \frac{u_s \rho}{A} \frac{dA}{ds} = 0 \quad , \tag{B.4}$$

$$u_s \rho \frac{du_s}{ds} = - \frac{dP}{ds} - \rho g_s \quad , \tag{B.5}$$

$$\rho u_s \frac{d\epsilon}{ds} = - \frac{dq_s}{ds} - P \frac{du_s}{ds} - \frac{(Pu_s + q_s)}{A} \frac{dA}{ds} - \mathcal{L} + S \quad , \tag{B.6}$$

$$q_s = K \frac{dT}{ds} \quad , \tag{B.7}$$

$$P = \frac{1}{u} \rho KT \quad , \tag{B.8}$$

where $A$ is the area of the flux tube $u$ is the mean particle mass, $K$ is the heat conductivity, $k$ is Boltzmann's constant, $T$ is the fluid temperature, $s$ measures distance along the flux tube and the subscript $s$ denotes the component of a vector along the flux tube. We have neglected transport of energy and momentum across field lines in writing equations (B.4)-(B.7). We wish to apply Equations (B.4)-(B.8) to the solar atmosphere. For this case we shall assume the plasma to be pure hydrogen except for computing the radiative losses. To account for radiative losses, we have assumed that it is reasonable to treat the solar atmosphere as optically thin (we discuss this assumption later). We have adopted the radiative loss function calculated by Raymond _et al._ (1976) as modified by Raymond (1976) to include radiative losses from Ar and neutral hydrogen excitation, but excluding radiative losses due to forbidden lines for temperatures below $T=10^6$ K . We have used the values of $u$ and $K$ derived by Moore and Fung (1972) for a pure hydrogen plasma. With these and the equation of state, we may eliminate the pressure. We choose as our dependent variables $q_s$ , $T$ , $u_s$ and $n$ , the number density of hydrogen nuclei and rewrite (B.4)-(B.7) in a form more convenient for numerical solution:

$$\frac{dT}{ds} = -\frac{q_s}{K} \tag{B.9}$$

$$\frac{dn}{ds} = \left(\frac{m_H}{(1+\chi)kT} - u_s^2\right)^{-1} \left\{ \frac{u_s^2 n}{A}\frac{dA}{ds} - \frac{dT}{ds}\left[\frac{(1+\chi)nk}{m_H} + \frac{nkT}{m_H}\frac{d\chi}{dT}\right] - ng_s \right\}, \tag{B.10}$$

$$\frac{dq_s}{ds} = -\mathcal{L} + S - \frac{3}{2}u_s nk\frac{dT}{ds}\left[(1+\chi) + (T-T_i)\frac{d\chi}{dT}\right] + (1+\chi)kTu_s\frac{dn}{ds} - q_s\frac{1}{A}\frac{dA}{ds}, \tag{B.11}$$

$$nu_sA = \text{constant} \tag{B.12}$$

where $m_H$ is the mass of a hydrogen atom, $\chi$ is the fraction of hydrogen nuclei that are ionized, and $T_i$ is the hydrogen ionization energy expressed as a temperature, $\sim 1.05 \times 10^5$ K. Equation (B.12) is the integral of Equation (B.4), we need only solve three first order ordinary differential equations to calculate the run of temperature and density in a flux tube.

We have written an assembly-language subroutine, to execute on IBM 360 or 370 series computers, to evaluate the quantities $\frac{dT}{ds}$ , $\frac{dn}{ds}$ and $\frac{dq_s}{ds}$ , given $A$ , $\frac{dA}{ds}$ , $g_s$ , $S$ , $n$ , $T$ and $q_s$ . This subroutine may be used with a standard library ordinary differential solver, or as we have done with one coded specially for this problem. The quantities $K$ , $\chi$ , $\frac{d\chi}{dt}$ , $\mathcal{L}$ , $A$ , $g_s$ , $\frac{dA}{ds}$ , and $S$ are tabulated as a function of $T$

and $s$, and the values for a particular $T$ or $s$ are computed by a cubic interpolation scheme similar to the one described in Appendix A.

The subroutine we have written to solve the coupled set of ordinary differential equations (B.9)-(B.12) uses an Adams-Bashforth-Moulton fourth order linear multistep integration scheme (see Isaacson and Keller 1966) with a fourth order Runge-Kutta scheme (with a smaller step size) to "start up" the linear multistep method and provide intermediate values when halving the step size. The routine returns the values of $T$, $u$, $q$ and $n$ at intervals from the starting point specified by the calling program and reduces the step size or increases it according to the requested accuracy. The pretabulated quantities are read in by the main program which also reads in starting values, calls the differential equation solver and writes out the results of the integration.

The downward heat flux in the corona above an active region is $\sim 5 \times 10^6$ (Noyes 1971). Since the thermal conductivity of the solar plasma is a strong function of temperature ($\propto T^{5/2}$), this heat flux must be largely radiated away above the low chromosphere. We have the choice of starting with our initial values where the heat flux is large (in the corona) and calculating the solutions to a region where the heat flux is small (the chromosphere), or proceeding in the reverse direction from the region where the heat flux is small. It is well known that the latter choice is preferable numerically (Acton 1970, Isaacson and Keller 1966). This is basically because the numerical calculation proceeding from the region of large heat flux to the region of small heat flux is not a "well posed" problem (Isaacson and Keller 1966) since a small relative change in the initial value of the heat flux can cause a large relative change in the final value. We therefore shall choose our starting point near the temperature minimum.

There are three major difficulties with starting the calculation below $3 \times 10^4$K. The first is that the atmosphere becomes optically thick and therefore the radiative losses cannot be calculated simply. Second, the radiative loss function calculated by Raymond is not tabulated below $10^4$K. Third, the approximation that the atmosphere is purely hydrogen breaks down as the fraction of ionized hydrogen becomes very small because the electron density (which appears in the expression for the radiative losses) is grossly underestimated by (A.4), since the major contribution to the electron density is from trace elements with low ionization potentials (e.g. Na). However, for the purposes of this work, we only need a model that represents the overall structure of the atmosphere reasonably well. This is particularly true since (cf. Chapter 3) the calculation of the heating of the cool dense portions of the atmosphere by the reverse current is not accurate after the first few tenths of a second due to the neglect of Coulomb collisions. We do not attempt a solution of the radiative transfer problem. We use a power law extrapolation of Raymond's (1976) radiative loss coefficient. We also use Equation (A.4) to find the electron density. The fact that the atmosphere is not optically thin is compensated for by the underestimate of the electron density. We have extrapolated Raymond's (1976) radiative loss function with a power law above and below the tabulated range ($T=10^4$-T $10^8$K). For the high temperatures above $T=10^8$K, this should be a reasonable approximation since the losses for these temperatures are almost completely due to thermal bremsstrahlung and therefore should vary as $\sim T^{1/2}$ ; however, these temperatures are not of importance in the present calculation. The power law extrapolation below $10^4$K is purely ad hoc, but the range over which extrapolated values are used is small ($\sim$ a factor of 2) and the calculation of radiative losses for these temperatures is at best approximate in any event. The resulting temperature and density profiles resemble the solar atmosphere in overall structure. Since the atmosphere varies from active region to active region, this should provide an adequate representation for the purposes of the calculations of Chapter 3.

To produce the model used (see Chapter 3), we integrate up from near the temperature minimum ($T=4200K$, $n=1.1025 \times 10^{16}$). The heat flux and velocity are taken to be zero at this point. No non-thermal energy input was included in the calculation. The resulting temperature, density and heat flux at the top of the model (corresponding to the injection point for the beam in Chapter 3) were $T=3 \times 10^6$, $n=1 \times 10^9$ cm$^{-3}$ and $F=6.30 \times 10^6$ erg cm$^{-2}$ s$^{-1}$, in reasonable agreement with the values given by Noyes (1971).

Listings of two main programs and several subroutines are provided for the sake of completeness. The first main program and associated subroutines produce the tables that are required for the cubic interpolation. The second main program reads in starting values for the solution of the coupled set of differential equations and writes out the results both as tables suitable for people to look at and (if desired) for machines to read. The subroutine ABMINT is the differential equation solver described above. The present version is in FORTRAN and is certainly adequate for the purpose of this work. An adaptation of the present main program to solve a boundary value problem rather than an initial value problem would (absent the wealth of Croesus) require this routine to be hand coded. The assembly language subroutine DIVF calculates the quantities needed by ABMINT to integrate the differential equations.

## TABULATION ROUTINES

```
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 G(820),DADS(820),SOR(820),ONEK(820),CHI(820),DCHI(820),
     .A(820),LUM(820),TST/Z441000000000000/,DT/Z431000000000000/,
     .SST/Z471000000000000/,DS/Z461000000000000/
      COMMON /PARAM/ FRAC,AMP,SCALE
 9001 FORMAT(10A8)
 5001 FORMAT(F13.6)
C
C     THIS PROGRAM CALCULATES SEMI-LOGARITHMIC INTERPOLATION
C     TABLES FOR A STEADY STATE MODEL OF THE PLASMA IN A MAGNETIC
C     FLUX TUBE UNDER THE INFLUENCE OF GRAVITY.  FOUR FUNCTIONS
C     OF TEMPERATURE AND FOUR FUNCTIONS OF S (DISTANCE ALONG THE
C     FLUX TUBE FROM THE SUN'S SURFACE) ARE TABULATED. THE
C     FUNCTIONS OF TEMPERATURE (T) ARE:
C
C            THE INVERSE OF THE THERMAL CONDUCTIVITY (ONEK)
C            THE RADIATIVE LOSS COEFFICIENT (LUM)
C            THE IONIZATION FRACTION (CHI)
C            THE DERIVATIVE OF THE IONIZATION FRACTION (DCHI)
C
C     THE FUNCTIONS OF S ARE:
C
C            THE FORCE OF GRAVITY ALONG THE TUBE (G)
C            THE AREA OF THE TUBE (A)
C            THE LOGARITHMIC DERIVATIVE OF THE AREA (DADS)
C            THE NON-THERMAL ENERGY INPUT (SOR)
C
C     THE TABULATION RANGE IN TEMPERATURE IS 4.096E3 - 6.71E7 (K).
C     THE TABULATION RANGE IN S IS 1.677E7 - 2.75E11 (CM).
C
C     THE TABLES ARE WRITTEN OUT TO FORTRAN LOGICAL UNIT 9 AND THE
C     PROGRAM READS IN 641 VALUES OF TEMPERATURE AND RADIATIVE
C     LOSS COFFICIENT (RAYMOND, PRIVATE COMMUNICATION) USED TO
C     TABULATE THE RADIATIVE LOSS COFFICIENT.
C
C     THE PROGRAM ALSO READS IN SEVERAL PARAMETERS THAT CHARACTERIZE
C     THE FLUX TUBE AND THE NON-THERMAL ENERGY INPUT:
C
C            FRAC:  THE AREA OF THE FLUX TUBE IS ((D+S)/D)**2 WHERE
C            D IS FRAC TIMES A SOLAR RADIUS.
C
C            AMP:  THE INTEGRAL OF THE NON-THERMAL ENERGY DEPOSITED IN
C            A FLUX TUBE OF CONSTANT AREA IS AMP (ERG PER CM**2 PER SEC).
C
C            SCALE:  THE FORM OF THE NON-THERMAL ENERGY INPUT IS
C            (COS((S*PI)/(2*SCALE)))**2 FOR S LESS THAN SCALE
C            AND ZERO FOR S GREATER THAN SCALE.  SCALE IS INPUT
C            IN SOLAR RADII (INPUT OF 1. MEANS SCALE IS ABOUT 7.E10 CM).
C
C
C     INITIALIZE TABLES:
C
      DO 1 I=1,820
        G(I)=0.D0
        DADS(I)=0.D0
        SOR(I)=0.D0
        A(I)=1.D0
        ONEK(I)=0.D0
        CHI(I)=0.D0
        DCHI(I)=0.D0
    1   LUM(I)=0.D0
C
C     READ IN PARAMETERS
C
      READ(5,5001)FRAC
```

```
      READ(5,5001)AMP
      READ(5,5001)SCALE
C
C     CALCULATE FUNCTIONS OF TEMPERATURE:
C
      DO 20 I=1,3
         T=TST-DT
         K=(I-1)*256+1
         DO 10 J=1,243
            CHI(K)=FCHI(T)
            DCHI(K)=FDCHI(T)
            ONEK(K)=FKAP(T)
            LUM(K)=FLUM(T)
            T=T+DT
   10       K=K+1
         TST=16.D0*TST
   20    DT=16.D0*DT
      T=TST-DT
      DO 30 K=769,820
         CHI(K)=FCHI(T)
         DCHI(K)=FDCHI(T)
         ONEK(K)=FKAP(T)
         LUM(K)=FLUM(T)
   30    T=T+DT
C
C     CALCULATE FUNCTIONS OF S:
C
      DO 50 I=1,3
         S=SST-DS
         K=(I-1)*256+1
         DO 40 J=1,243
            G(K)=FG(S)
            A(K)=FA(S)
            DADS(K)=FDADS(S)
            SOR(K)=FSOR(S)
            S=S+DS
   40       K=K+1
         SST=16.D0*SST
   50    DS=16.D0*DS
      S=SST-DS
      DO 60 K=769,820
         G(K)=FG(S)
         A(K)=FA(S)
         DADS(K)=FDADS(S)
         SOR(K)=FSOR(S)
   60    S=S+DS
C
C     WRITE OUT TABLES:
C
      WRITE(9,9001)G
      WRITE(9,9001)DADS
      WRITE(9,9001)SOR
      WRITE(9,9001)A
      WRITE(9,9001)ONEK
      WRITE(9,9001)LUM
      WRITE(9,9001)CHI
      WRITE(9,9001)DCHI
      STOP
      END
      REAL FUNCTION FCHI*8(T)
C
C     THIS FUNCTION CALCULATES THE IONIZATION FRACTION AS A FUNCTION
C     OF THE TEMPERATURE (T).  THE IONIZATION FRACTION (FCHI) IS
C     DEFINED AS NE/(NH+NP) WHERE NE IS THE NUMBER DENSITY OF
C     ELECTRONS, AND NH AND NP ARE THE NUMBER DENSITIES OF HYDROGEN
C     ATOMS AND PROTONS RESPECTIVELY.  SEE MOORE AND FUNG, SOLAR
C     PHYSICS 23 (1972),78-102 FOR FORMULAE.
C
      IMPLICIT REAL*8 (A-H,O-Z)
```

```
      DATA ONE3/ZC05555555555555/
      COMMON BETA,EBETA,B13,TEMP1,TEMP2,TCHI,D
      BETA=1.58D5/T
      EBETA=DEXP(BETA)
      B13=BETA**ONE3
      TEMP1=0.4288D0+0.5D0*DLOG(BETA)+.4698D0*B13
      TEMP2=2.22D-6*BETA*TEMP1*EBETA
      TCHI=1.D0/(1.D0+TEMP2)
      FCHI=TCHI
      RETURN
      END
      REAL FUNCTION FDCHI*8(T)
C
C     THIS FUNCTION CALCULATES THE DERIVATIVE OF THE IONIZATION FRACTION
C     (D DHI / DT) AS A FUNCTION OF TEMPERATURE (T).  SEE FUNCTION FCHI.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON BETA,EBETA,B13,TEMP1,TEMPC,TCHI,D
      FDCHI=1.406D-11*TCHI*TCHI*BETA*BETA*EBETA*((1.D0+BETA)*TEMP1
     .+ (.5D0-.1566D0*B13))
      RETURN
      END
      REAL FUNCTION FKAP*8(T)
C
C     THIS FUNCTION CALCULATES THE INVERSE OF THE TOTAL THERMAL
C     CONDUCTIVITY AS A FUNCTION OF TEMPERATURE.  THE DEPENDENCE OF THE
C     CONDUCTIVITY ON THE "COULOMB LOGARITHM" IS APPROXIMATED IN A
C     MANNER SIMILAR TO MOORE AND FUNG, SOLAR PHYSICS 23 (1972), 78-102.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON BETA,EBETA,B13,TEMP1,TEMPC,TCHI,D
      REAL*8 P0/0.D0/,CL/0.D0/,CK1/0.D0/,RKAY/1.38062D-16/,
     .MPROT/1.67352D-24/,ESU/4.80325D-10/,PI/Z413243F6A8885A30/
      IF(P0.NE.0.D0)GOTO 10
      P0=1.D5*1.D10*(2.D0*RKAY)
      CL=(3.D0*RKAY**2)/(DSQRT(2.D0*PI*P0)*ESU**3)
      CK1=(9.D0*RKAY*DSQRT(RKAY))/(4.D0*DSQRT(MPROT))
   10 CLAM=(T*T*CL)*DSQRT((1.D0+TCHI)/(2.D0*TCHI))
      T12=DSQRT(T)
      IF(T.GT.4.2D5)CLAM=CLAM*6.48074D2/T12
      TEMPK=(CK1*T)/(9.12D-14+7.95D-11/(TEMPC*T12))
      FKAP=1.D0/(TEMPK+(1.89D-5*T*T*T12)/DLOG(CLAM))
      RETURN
      END
      REAL FUNCTION FLUM*8(T)
C
C     THIS FUNCTION CALCULATES THE RADIATIVE LOSS COEFFICIENT
C     SUCH THAT THE RADIATIVE LOSSES FROM AN OPTICALLY THIN
C     PLASMA OF SOLAR ABUNDANCESARE FLUM*(NE**2) WHERE NE IS
C     THE ELECTRON NUMBER DENSITY.  THE CALCULATION OF THE
C     RADIATIVE LOSS COEFFICIENT IS RAYMOND'S (PRIVATE COMM.)
C     IMPROVEMENT OF THE CALCULATIONS OF RAYMOND, COX AND SMITH
C     AP. J. 204 (1976), 290-292.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 T,L,TD(641),LD(641),LOGT,ERR,FINT(10),XDIF(10),WRK(10)
      REAL*4 RTD(641),RLD(641)
      LOGICAL SORT/.FALSE./,EXTRAP/.FALSE./,FIRST/.TRUE./
      EQUIVALENCE (TD(321),RTD(1)),(LD(321),RLD(1))
 8001 FORMAT(20A4)
      IF(FIRST)GOTO 100
  110 LOGT=DLOG10(T)
      IF(LOGT.LT.TD(1))GOTO 200
      IF(LOGT.GT.TD(NRADPT))GOTO 300
      ERR=-1.D0
      CALL AITKEN(L,LOGT,10,ERR,TD,LD,NRADPT,SORT,EXTRAP,FINT,XDIF,WRK,
     .&10,&400,&400)
   10 FLUM=10.D0**L
      RETURN
```

```
  100 NRADPT=641
C
C     CALCULATING LEAST SQUARE FITS FOR POWER LAW EXTENSION
C     OF CALCULATED RADIATIVE LOSS COEFFICIENT BEYOND TABULATED
C     RANGE. ONLY DO ON FIRST CALL.
C
      READ(8,8001)RTD
      READ(8,8001)RLD
      DO 15 I=1,NRADPT
      TD(I)=DBLE(RTD(I))
   15 LD(I)=DBLE(RLD(I))
      A1=0.D0
      B1=0.D0
      TEMP1=0.D0
      TEMP2=0.D0
      DO 20 I=2,10
      A1=A1+LD(I)
      B1=B1+TD(I)
      TEMP1=TEMP1+TD(I)*TD(I)
   20 TEMP2=TEMP2+TD(I)*LD(I)
      B1=(TEMP2-TD(1)*A1-LD(1)*(B1-9.D0*TD(1)))/
     .(TEMP1-2.D0*TD(1)*B1+9.D0*TD(1)*TD(1))
      A1=LD(1)-B1*TD(1)
      A2=0.D0
      B2=0.D0
      TEMP1=0.D0
      TEMP2=0.D0
      DO 30 I=636,640
      A2=A2+LD(I)
      B2=B2+TD(I)
      TEMP1=TEMP1+TD(I)*TD(I)
   30 TEMP2=TEMP2+TD(I)*LD(I)
      B2=(TEMP2-TD(641)*A2-LD(641)*(B2-5.D0*TD(641)))/
     .(TEMP1-2.D0*TD(641)*B2+5.D0*TD(641))
      A2=LD(641)-B2*TD(641)
      FIRST=.FALSE.
      GOTO 110
  200 FLUM=10.D0**(A1+B1*LOGT)
      RETURN
  300 FLUM=10.D0**(A2+B2*LOGT)
      RETURN
  400 WRITE(6,6001)
 6001 FORMAT(1H ,'OOPS - WE SHOULD NOT BE HERE')
      STOP
      END
      REAL FUNCTION FG*8(S)
C
C     THIS FUNCTION CALCULATES THE FORCE OF GRAVITY ALONG THE
C     FLUX TUBE AS A FUNCTION OF S, THE DISTANCE ABOVE THE
C     SOLAR SURFACE.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 RSUN/6.9599D10/,G/6.67D-8/,MSUN/1.989D33/
      LOGICAL NOT1ST/.FALSE./
      IF(NOT1ST)GOTO 10
      GM=MSUN*G
      NOT1ST=.TRUE.
   10 R=(RSUN+S)
      FG=GM/(R**2)
      RETURN
      END
      REAL FUNCTION FA*8(S)
C
C     THIS FUNCTION CALCULATES THE AREA OF THE FLUX TUBE
C     AS A FUNCTION OF S, THE DISTANCE ABOVE THE SURFACE OF
C     THE SUN.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON BETA,EBETA,B13,TEMP1,TEMPC,TCHI,D
```

```
      REAL*8 RSUN/6.9599D10/,A0/1.D0/
      LOGICAL NOT1ST/.FALSE./
      COMMON /PARAM/ FRAC,AMP,SCALE
      IF(NOT1ST)GOTO 10
      D=FRAC*RSUN
      R=S+D
      AR=R**2
      AR=A0/AR
      FA=AR*(R**2)
      NOT1ST=.TRUE.
      RETURN
   10 R=D+S
      FA=AR*(R**2)
      RETURN
      END
      REAL FUNCTION FDADS*8(S)
C
C     THIS FUNCTION CALCULATES THE LOGARITHMIC DERIVATIVE
C     OF THE AREA AS A FUNCITON OF S, THE DISTANCE ABOVE THE
C     SURFACE OF THE SUN.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON BETA,EBETA,B13,TEMP1,TEMPC,TCHI,D
      FDADS=2.D0/(D+S)
      RETURN
      END
      REAL FUNCTION FSOR*8(S)
C
C     THIS FUNCTION CALCULATES THE (AD HOC) NON-THERMAL
C     ENERGY INPUT INTO THE SOLAR PLASMA AS A FUNCTION OF S,
C      THE DISTANCE ABOVE THE SUN'S SURFACE.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 RSUN/6.9599D10/,
     .PIBY2/Z411921FB54442D18/
      LOGICAL NOT1ST/.FALSE./
      COMMON /PARAM/ FRAC,AMP,SCALE
      IF(NOT1ST)GOTO 10
      SCALE=SCALE*RSUN
      ARG=1.D0/SCALE
      AMP=AMP*ARG
      AMP=AMP+AMP
      ARG=ARG*PIBY2
      S0=S
      NOT1ST=.TRUE.
   10 SR=S-S0
      IF(SR.GT.SCALE)GOTO 20
      C=DCOS(ARG*SR)
      FSOR=AMP*C*C
      RETURN
   20 FSOR=0.D0
      RETURN
      END
      SUBROUTINE AITKEN(F,X,M,ERR,XTAB,FTAB,N,SORT,EXTRAP,FINT,
     .XDIF,WRK,*,*,*)
C
C     SUBROUTINE AITKEN INTERPOLATES TO FIND THE VALUE OF THE FUNCTION
C     (F) AT THE POINT X.  IF THE ROUTINE DOES NOT ACHIEVE THE DESIRED
C     RELATIVE ERROR (ERR) USING M POINTS OR IF ROUND OFF ERROR APPEARS
C     TO BE PRESENT, THE ROUTINE RETURNS THE CURRENT ERROR ESTIMATE IN
C     ERR, RETURNING TO THE MAIN PROGRAM AT THE FIRST STATEMENT NUMBER
C     IN THE ARGUMENT LIST.  THE ROUTINE REQUIRES THE TABULATED VALUES
C     IN FTAB TO BE IN ORDER OF INCREASING VALUE OF X (IN XTAB).  IF
C     SORT IS TRUE ON ENTRY, BOTH TABLES ARE SORTED (SEE NOTE).  IF THE
C     VALUE OF X IS OUTSIDE THE RANGE OF THE TABLES SUPPLIED, THE
C     ROUTINE RETURNS TO THE SECOND STATEMENT NUMBER IN THE ARGUMENT
C     LIST - UNLESS EXTRAP IS TRUE.  IF THE ROUTINE DISCOVERS TWO
C     IDENTICAL VALUES OF X IN XTAB, THE ROUTINE RETURNS TO THE THIRD
C     STATEMENT NUMBER IN THE ARGUMENT LIST.
```

```
C
C     IF M IS GREATER THAN N OR LESS THAN 2, IT IS SET TO 10.
C
C     IF ERR IS LESS THAN 16**-5, IT IS SET TO 16**-5.
C
C     ARGUMENTS (OTHER THAN STATEMENT NUMBERS):
C
C         F  INTERPOLATED VALUE OF FUNCTION AT X (REAL - OUTPUT)
C
C         X  VALUE OF INDEPENT VARIABLE (REAL - INPUT)
C
C         M  LARGEST NUMBER OF DATA POINTS TO BE USED (INTEGER - INPUT)
C
C         ERR  REQUESTED RELATIVE ERROR (REAL - INPUT)
C
C         XTAB  TABLE OF X VALUES AT WHICH F(X) IS TABULATED
C               (REAL ARRAY - INPUT)
C
C         FTAB  TABLE OF F(X) AT THE CORRESPONDING POINTS IN XTAB
C               (REAL ARRAY - INPUT)
C
C         N  THE LENGTH OF TABLES XTAB AND FTAB (INTEGER - INPUT)
C
C         SORT  DETERMINES WHETHER OR NOT THE INTERNAL SORING ROUTINE
C               IS TO BE USED (LOGICAL - INPUT/OUTPUT)
C
C         EXTRAP  DETERMINES WHETHER OR NOT EXTRAPOLATION OUTISDE THE
C                 RANGE OF THE TABLES IS ALLOWED(LOGICAL -INPUT)
C
C         FINT  ARRAY OF SUCESSIVE INTERPOLANTS - WORKING ARRAY
C               (REAL ARRAY DIMENSION > OR = M)
C
C         XDIF  ARRAY OF DIFFERENCES BETWEEN THE POINTS AT WHICH F(X)
C               IS TABULATED AND X - WORKING ARRAY (REAL ARRAY
C               DIMENSION > OR = M)
C
C         WRK  WORKING ARRAY FOR CURRENT LEVEL OF INTERPOLATION
C              (REAL ARRAY DIMENSION > OR = M)
C
C     INTERNAL VARIABLES:
C
C         TEMP  TEMPORARY STARAGE LOCATION FOR INTERMEDIATE RESULTS
C
C         FDIFF1   PREVIOUS ABSOLUTE RELATIVE DIFFERENCE BETWEEN
C                  INTERPOLANTS - COMPARED WITH FDIFF2 TO CHECK FOR
C                  CONVERGENCE (ROUND-OFF ERROR INDICATOR)
C
C         FDIFF2   PRESENT ABSOLUTE RELATIVE DIFFERENCE BETWEEN
C                  INTERPOLANTS - USED TO CHECK FOR CONVERGENCE AT
C                  CURRENT LEVEL (ALSO SEE FDIFF1 ABOVE)
C
C         DIFFMAX  LARGEST REPRESENTABLE FLOATING POINT NUMBER (IBM 360)
C
C         IUP  USED AS POINTER IN SORT AND INTERPOLATION
C
C         IMID  USED AS POINTER IN SORT
C
C         IDN  USED AS POINTER IN SORT AND INTERPOLATION
C
C         XUPDIF  DIFFERENCE BETWEEN X AND CLOSEST UNUSED LARGER VALUE
C                 IN XTAB
C
C         XDNDIF  DIFFERENCE BETWEEN X AND CLOSEST UNUSED SMALLER VALUE
C                 IN XTAB
C
C         LEVEL  CURRENT LEVEL OF AITKEN TRIANGULAR SCHEME
C
C         ISTEP  COUNTER FOR INTERMEDIATE INTERPOLANT LOOP
C
```

```
C          DONE  LOGICAL FLAG TO INDICATE CURRENT LEVEL OF SHELL SORT
C                IS COMPLETE
C
C          IDISP  CURRENT EXCHANGE INTERVAL IN SHELL SORT
C
C          ILAST  N MINUS IDISP - UPPER LIMIT FOR SORT DO LOOP
C
C          I  COUNTER IN SORT DO LOOP
C
C      REMARKS:
C
C          THE ROUTINE AS PRESENTLY WRITTEN WILL NOT WORK IN WATFIV.  TO
C          MAKE THE ROUTINE COMPATABLE WITH WATFIV, THREE CHANGES MUST BE
C          MADE.  FIRST, THE ARRAYS FINT, XDIF AND WRK SHOULD HAVE
C          DIMENSION M AND THE ARRAYS XTAB AND FTAB SHOULD HAVE THE
C          DIMENSION N.  SECOND, THE VARIABLES M AND N SHOULD BE REMOVED
C          FROM THE INTEGER DECLARATION STATEMENT.  THIRD, THE STATEMENT
C          WHICH CHANGES M TO 10 IF CERTAIN CONDITIONS ARE MET SHOULD BE
C          DELETED.
C
C      NOTE:
C
C          SORT METHOD USED IS SHELL SORT - THIS METHOD MAY BE VERY
C          INEFFICIENT WHEN XTAB IS PARTIALLY SORTED.
C
C
C      DECLARE VARIABLES
C
      REAL F,X,ERR,XTAB(1),FTAB(1),FINT(1),XDIF(1),WRK(1),EPS,XUPDIF,
     .XDNDIF,FDIFF,FDIFF2,DFIMAX,TEMP
      INTEGER M,N,ISTEP,ILAST,LEVEL,IDISP,IUP,IDN,IMID,I
      LOGICAL SORT,EXTRAP,DONE
C
C      INITIALIZE VARIABLES
C
      DATA EPS/Z3C100000/,DIFMAX/Z7FFFFFFF/
C
C      CHECK TO SEE IF M > N OR IF M < 2, IF SO SET M TO 10
C      (THIS CARD MUST BE REMOVED FOR WATFIV EXECUTION AND THE
C       WORKING ARRAYS DIMENSIONED TO M)
C
      IF(M.LT.2.OR.M.GT.N)M=10
C
C      CHECK TO SEE IF ERR < 16**-5 IF SO SET IT TO 16**-5
C
      IF(ERR.LT.EPS)ERR=EPS
C
C      CHECK TO SEE IF TABLES ARE TO BE SORTED - IF NOT GO AROUND SORT
C      SECTION.
C
      IF(.NOT.SORT)GOTO 200
C
C      ****  SORTING SECTION BEGIN
C
      IDISP=N
  101 IDISP=(IDISP+1)/2
      ILAST=N-IDISP
  102 DONE=.TRUE.
      DO 103 I=1,ILAST
      IF(XTAB(I).LT.XTAB(I+IDISP))GOTO 103
      IF(XTAB(I).EQ.XTAB(I+IDISP))RETURN 3
      TEMP=XTAB(I)
      XTAB(I)=XTAB(I+IDISP)
      XTAB(I+IDISP)=TEMP
      TEMP=FTAB(I)
      FTAB(I)=FTAB(I+IDISP)
      FTAB(I+IDISP)=TEMP
      DONE=.FALSE.
  103 CONTINUE
```

```
      IF(.NOT.DONE)GOTO 102
      IF(IDISP.GT.1)GOTO 101
C
C     ****  SORTING SECTION END
C
  200 CONTINUE
C
C     CHECK TO SEE IF X IS WITHIN RANGE OF TABLE - IF NOT AND IF EXTRAP
C     IS FALSE RETURN TO SECOND STATEMENT IN ARGUMENT LIST
C
      IF(X.GE.XTAB(1))GOTO 201
C
C     X IS BELOW LOWEST X VALUE IN XTAB - EXIT UNLESS EXTRAP IS TRUE
C
      IF(.NOT.EXTRAP)RETURN 2
C
C     EXTRAP IS TRUE - SET UP POINTERS AND GO TO AITKEN
C     INTERPOLATION SECTION
C
      IUP=2
      IDN=1
      GOTO 400
  201 CONTINUE
C
C     CHECK TO SEE IF X IS LARGER THAN LARGEST X VALUE IN XTAB - IF NOT
C     BRANCH TO SEARCH SECTION
C
      IF(X.LE.XTAB(N))GOTO 300
C
C     X IS ABOVE HIGHEST X VALUE IN XTAB - EXIT UNLESS EXTRAP IS TRUE
C
      IF(.NOT.EXTRAP)RETURN 2
C
C     EXTRAP IS TRUE - SET UP POINTERS AND GO TO AITKEN
C     INTERPOLATION SECTION
C
      IUP=N
      IDN=N-1
      GOTO 400
C
C     SEARCH SECTION - FIND XTAB VALUES THAT BRACKET X - USE BISECTION
C
  300 CONTINUE
C
C     SET UP POINTERS FOR BISECTION
C
      IUP=N
      IMID=N/2
      IDN=1
C
C     CHECK TO SEE WHICH SIDE OF EXTAB(IMID) X IS ON AND UPDATE IUP,
C     IMID AND IDN - WHEN NEW IMID EQUALS IDN WE ARE DONE
C
  301 IF(X.GT.XTAB(IMID))GOTO 302
C
C     X LE XTAB(IMID) SO IUP<=IMID & IMID<=(IUP+IDN)/2
C
      IUP=IMID
      IMID=(IUP+IDN)/2
C
C     IF IMID > IDN WE AREN'T DONE YET - GO BACK AND CHECK AGAIN
C     OTHERWISE GO TO AITKEN INTERPOLATION SECTION
C
      IF(IMID.GT.IDN)GOTO 301
      GOTO 400
  302 CONTINUE
C
C     X > XTAB(IMID) SO IDN<=IMID & IMID<=(IUP+IDN)/2
C
```

```
      IDN=IMID
      IMID=(IUP+IDN)/2
C
C     IF IMID > IDN WE AREN'T DONE YET - GO BACK AND CHECK AGAIN
C     OTHERWISE ENTER AITKEN INTERPOLATION SECTION
C
      IF(IMID.GT.IDN)GOTO 301
C
C     END OF SEARCH SECTION
C
C     AITKEN INTERPOLATION SECTION
C
  400 CONTINUE
C
C     IUP AND IDN POINT TO FIRST TWO FUNCTION VALUES USED IN
C     INTERPOLATION - INITIALIZE VARIABLES
C
      FDIFF2=DIFMAX
      XDNDIF=XTAB(IUP)-X
      XUPDIF=XTAB(IDN)-X
C
C     START AITKEN INTERPOLATION
C
      DO 401 LEVEL=1,M
C
C        DECIDE WHICH OF THE TWO TABLE VALUES POINTED TO BY IUP AND IDN
C        IS TO BE USED NEXT - THE ONE WITH XTAB CLOSER TO X
C
         IF(ABS(XUPDIF).GT.ABS(XDNDIF))GOTO 402
C
C        WE WILL USE IUP - PUT INFORMATION IN WORKING ARRAYS
C
         WRK(1)=FTAB(IUP)
         XDIF(LEVEL)=XUPDIF
C
C        CHECK TO SEE IF WE JUST USED THE LARGEST VALUE OF X IN XTAB
C        IF SO GO TO 403 AND DO FIX UP - IF NOT UPDATE IUP AND XUPDIF
C
         IF(IUP.GE.N)GOTO 403
         IUP=IUP+1
         XUPDIF=XTAB(IUP)-X
C
C        BRANCH AROUND CODE TO INTERPOLATION LOOP FOR THIS LEVEL
C
         GOTO 404
C
C        FIX UP FOR USE OF LARGEST X IS TO SET XUPDIF TO LARGEST
C        REPRESENTABLE FLOATING POINT NUMBER
C
  403    XUPDIF=DIFMAX
C
C        BRANCH AROUND CODE TO INTERPOLATION LOOP FOR THIS LEVEL
C
         GOTO 404
C
C        WE WILL USE IDN - PUT INFORMATION IN WORKING ARRAYS
C
  402    WRK(1)=FTAB(IDN)
         XDIF(LEVEL)=XDNDIF
C
C        CHECK TO SEE IF WE USED THE SMALLEST VALUE OF X IN X IN XTAB
C        IF SO GO TO 405 AND DO FIX UP - IF NOT UPDATE IDN AND XDNDIF
C
         IF(IDN.EQ.1)GOTO 405
         IDN=IDN-1
         XDNDIF=XTAB(IDN)-X
C
C        BRANCH AROUND CODE TO INTERPOLATION LOOP FOR THIS LEVEL
C
```

```
      GOTO 404
C
C     FIX UP FOR USE OF SMALLEST X IS TO SET XDNDIF TO LARGEST
C     REPRESENTABLE FLOATING POINT NUMBER
C
 405  XDNDIF=DIFMAX
C
C     SKIP INTERPOLATION CALCULATION IF LEVEL IS 1
C
 404  IF(LEVEL.LE.1)GOTO 406
C
C        AITKEN INTERPOLATION LOOP
C
         DO 407 ISTEP=2,LEVEL
         TEMP=XDIF(LEVEL)-XDIF(ISTEP-1)
C
C        CHECK TO SEE IF WE ARE GOING TO DIVIDE BY 0 IF SO RETURN
C        TO THIRD STATEMENT NUMBER IN ARGUMENT LIST
C
         IF(TEMP.EQ.0.)RETURN 3
C
C        CALCULATE INTERMEDIATE INTERPOLANTS
C
 407     WRK(ISTEP)=(FINT(ISTEP-1)*XDIF(LEVEL) -
     1              WRK(ISTEP-1)*XDIF(ISTEP-1))/TEMP
C
C     ENTER INTERPOLANT IN FINT
C
 406  FINT(LEVEL)=WRK(LEVEL)
C
C     SKIP CHECK FOR CONVERGENCE FOR LEVEL LESS THAN 4
C
      IF(LEVEL.LT.4)GOTO 401
C
C     CHECK FOR CONVERGENCE AT THIS LEVEL - IF SO BRANCH OUT
C
      FDIFF2=2.*ABS((FINT(LEVEL)-FINT(LEVEL-1))/
     1      (FINT(LEVEL)+FINT(LEVEL-1)))
      IF(FDIFF2.LT.ERR)GOTO 408
C
C     SKIP ROUND OFF ERROR CHECK FOR LEVEL LESS THAN 6
C
      IF(LEVEL.LT.6)GOTO 401
C
C     IF INTERPOLANTS ARE NOT CONVERGING - EXIT
C
      IF(FDIFF2.GT.FIDFF1)GOTO 501
C
C     UPDATE FDIFF1 AND CONTINUE
 401  FDIFF1=FDIFF2
C
C   IF INTERPOLATED TO LEVEL=M WITHOUT CONVERGENCENCE - EXIT
C
    GOTO 501
C
C   SET F EQUAL TO FINT(LEVEL) AND RETURN
C
 408 F=FINT(LEVEL)
    RETURN
C
C   TERMINATIONS DUE TO LACK OF CONVERGENCE OR ROUND OFF ERROR
C
 501 LEVEL=LEVEL-1
    ERR=FIDFF1
    F=FINT(LEVEL)
    RETURN 1
    END
```

STEADY STATE ATMOSPHERE MODEL MAIN ROUTINE

```
      IMPLICIT REAL*8 (A-H,O-Z)
C
C     THIS PROGRAM CALCULATES THE RUN OF TEMPERATURE, DENSITY
C     HEAT FLUX AND VELOCITY IN AN INDIVIDUAL FLUX TUBE.
C     THE PROGRAM READS IN PARAMETERS THAT CONTROL THE NUMBER OF
C     SETS OF TABLES READ IN (NTAB), AND THE INDEPENDENT VARIABLE
C     THAT CONTROLS THE FREQUENCY OF TABULATION (ITEST).  FOR EACH
C     SET OF TABLES THE PROGRAM READS IN THE NUMBER OF DIFFERENT INITIAL
C     CONDITIONS FOR WHICH THE INTEGRATION IS TO BE PERFORMED (NRUN)
C     AND A VARIABLE THAT CONTROLS WHETHER OR NOT THE RESULTS OF
C     THE INTEGRATION ARE ONLY PRINTED OUT OR BOTH PRINTED OUT
C     AND WRITTEN OUT IN A FORMAT SUITABLE FOR REREADING
C     BY ANOTHER PROGRAM (NOUT).  IF NOUT IS LESS THAN 1, THEN THE
C     RESULTS ARE ONLY PRINTED. IF NOUT IS GREATER THAN OR EQUAL TO
C     1, THEN THE RESULTS OF THE INTEGRATION ARE BOTH PRINTED OUT AND
C     WRITTEN OUT TO LOGICAL UNIT 10.
C
C     THE PROGRAM READS IN 4 TABULATED FUNCTIONS OF DISTANCE AND
C     4 TABULATED FUNCTIONS OF TEMPERATURE:
C
C     FUNCTIONS OF S:
C
C            G  THE FORCE OF GRAVITY ALONG THE TUBE
C
C            DA  THE LOGARITHMIC DERIVATIVE OF THE AREA OF THE TUBE
C                WITH RESPECT TO DISTANCE ALONG THE TUBE (1/A DA/DS)
C
C            SO  A PHENOMENOLOGICAL NON-THERMAL HEAT SOURCE
C
C            A  THE AREA OF THE FLUX TUBE
C
C     FUNCTIONS OF T:
C
C            OK  INVERSE OF THE THERMAL CONDUCTIVITY
C
C            LU  THE RADIATIVE LOSS FUNCTION
C
C            CH  THE FRACTION OF HYDROGEN NUCLEI THAT ARE IONIZED
C
C            DC  DERIVATIVE OF THE FRACITIONAL IONIZATION (CH)
C
C     FOR EACH RUN WITH A SET OF TABLES, THE PROGRAM READS IN SO, THE
C     STARTING DISTANCE, DSO THE INITIAL STEP SIZE, PRCT, THE
C     MULTIPLICATIVE FACTOR BY WHICH THE INDEPENDENT VARIABLE SELECTED
C     BY ITEST IS ALLOWED TO CHANGE BETWEEN TABULATION POINTS, THE
C     INITIAL TEMPERATURE TO, INITIAL DENSITY NO, INITIAL HEAT FLUX QO,
C     INITIAL VELOCITY UO, TSTOP, THE TEMPERATURE AT WHICH THE
C     INTEGRATION WILL STOP, SSTOP, THE DISTANCE AT WHICH THE
C     INTEGRATION WILL STOP, EPS, THE MAXIMUM RELATIVE ERROR IN AN
C     INDEPENDENT VARIABLE ALLOWED PER DSO, AND MSTOP, THE MACH NUMBER
C     AT WHICH THE INTEGRATION WILL STOP.  IF THE INITIAL HEAT FLUX READ
C     IN IS GREATER THAN 10 TO THE 50TH (A VERY UNPHYSICAL VALUE) THE
C     INITIAL HEAT FLUX IS DETERMINED BY THE CONDITION THAT THE NET
C     ENERGY FLUX IS ZERO AT THE STARTING POINT.
C
C     THE CURRENT VERSION INTERPOLATES THE RESULTS OF THE
C     INTEGRATION TO PRINT OUT VALUES OF DISTANCE TEMPERATURE,
C     DENSITY, HEAT FLUX, VELOCITY, PRESSURE AND A QUANTITY WHICH
C     CAN BE INFERRED FROM EUV OBSERVATIONS (P**2 KAPPA/Q WHERE KAPPA
C     IS THE THERMAL CONDUCTIVITY) AT VALUES OF THE TEMPERATURE
C     INITIALIZED IN THE ARRAY TMPOUT.  IN ADDITION THE INITIAL AND
C     FINAL POINTS OF THE INTEGRATION ARE PRINTED OUT.  IF THE
C     RESULTS ARE TO BE WRITTEN TO LOGICAL UNIT 10 ALL THE TABULATED
C     RESULTS ARE PRINTED AS CALCULATED BY ABMINT.
C
```

```
C
C      DECLARE AND INITIALIZE ARRAYS AND VARIABLES
C
       REAL*8 EPS,TSTOP,SSTOP,A0,U0,N0,T0,Q0,S0,S,DS0,DS,PRCT,
      .P,YPASS(4),KAY/Z339F2CB600000000/,
      .MSTOP
       REAL*8 G(819),DA(819),SO(819),A(819),OK(819),LU(819),CH(819),
      .DC(819),YTAB(5,2048),TAB(8),GRAV.DADS,SOR,AR,OKAP,LAM,CHI,DCHI
       INTEGER*4 NRUN,NTAB,IRUN,ITAB,I,J
       EQUIVALENCE (TAB(1),GRAV),(TAB(2),DADS),(TAB(3),SOR),(TAB(4),AR),
      .(TAB(5),OKAP),(TAB(6),LAM),(TAB(7),CHI),(TAB(8),DCHI)
       EXTERNAL DIVF
       REAL*8 SPRIN,SBOTT/Z474143E000000000/,TMPOUT(40)/
      .1.D3,1.5D3,2.D3,3.D3,4.D3,5.D3,6.D3,7.D3,8.D3,9.D3,
      .1.D4,1.5D4,2.D4,3.D4,4.D4,5.D4,6.D4,7.D4,8.D4,9.D4,
      .1.D5,1.5D5,2.D5,3.D5,4.D5,5.D5,6.D5,7.D5,8.D5,9.D5,
      .1.D6,1.5D6,2.D6,3.D6,4.D6,5.D6,6.D6,7.D6,8.D6,9.D6/
  5001 FORMAT(2I5)
  5002 FORMAT(3Z16)
  5003 FORMAT(4D13.6)
  6001 FORMAT(1H ,T4,'S(CM)',T19,'TEMP',T34,'N',T49,'Q(CGS)',
      .T64,'U(CGS)',T79,'P(CGS)',T94,'LUM',/)
  6002 FORMAT(7(1PD15.6))
  6004 FORMAT(1H1,T4,'S(CM)',T19,'TEMP',T34,'N',T49,'Q(CGS)',
      .T64,'U(CGS)',T79,'P(CGS)',T94,'LUM',/)
  9004 FORMAT(10A8)
C
C      CALL DIVINT - PASS BASE ADDRESSES OF INTERPOLATION TABLES
C      TO DIVF
C
       CALL DIVINT(G,DA,SO,A,OK,LU,CH,DC)
C
C      READ IN NUMBER OF SETS OF TABLES AND INDEX OF INDEPENDENT
C      VARIABLE THAT CONTROLS TABULATION FREQUENCY
C
       READ(5,5001)NTAB,ITEST
       PRCT=1.05D0
C
C      READ IN INTERPOLATION TABLES
C
       DO 999 ITAB=1,NTAB
       READ(5,5001)NRUN,NOUT
       READ(9,9004)G
       READ(9,9004)DA
       READ(9,9004)SO
       READ(9,9004)A
       READ(9,9004)OK
       READ(9,9004)LU
       READ(9,9004)CH
       READ(9,9004)DC
C
C      READ IN INITIAL CONDITIONS FOR RUNS
C
       DO 99 IRUN=1,NRUN
       READ(8,5002)S0,DS0,PRCT
       READ(8,5003)T0,N0,Q0,U0
       READ(8,5003)TSTOP,SSTOP,EPS,MSTOP
C
C      INITIALIZE A0 AND Q0 IF NECESSARY
C
       YPASS(1)=T0
       YPASS(2)=N0
       CALL DIVF(S0,YPASS,TAB)
       A0=AR*N0*U0
       I1=1
       IF(Q0.LE.1.E50)GOTO 5
       Q0=-.5D0*U0*N0*(U0*U0*1.67352D-24+5.D0*KAY*T0*(1.+CHI))
     5 YPASS(3)=Q0
       YPASS(4)=U0
```

```
      S=SO
      DS=DSO
      NMAX=2048
C
C     CALL ABMINT TO INTEGRATE EQUATIONS
C
      CALL ABMINT(S,YPASS,DIVF,DS,EPS,TSTOP,SSTOP,MSTOP,YTAB(1,I1),
     .AO,ITEST,PRCT,NMAX)
      I1=I1+NMAX-1
   20 CONTINUE
C
C     PRINT OUT RESULTS
C
C             WE LOOK FOR VALUES OF TEMPERATURE THAT BRACKET VALUES
C             OF TEMPERATURE IN TMPOUT AND INTERPOLATE.  WE ALSO
C             DO OUR OWN PAGINATION.
C
      WRITE(6,6004)
      ITEMP=0
      ILINE=1
      I=2
      SPRIN=YTAB(5,1)
      CALL DIVF(SPRIN,YTAB(1,1),TAB)
      P=YTAB(2,1)*(1.D0+CHI)*YTAB(1,1)
      RADOUT=1.D50
      IF(YTAB(3,1).EQ.0.D0)GOTO 30
      RADOUT=-(P*P)/(OKAP*YTAB(3,1))
   30 P=KAY*P
      SPRIN=SPRIN-SBOTT
      WRITE(6,6002)SPRIN,YTAB(1,1),YTAB(2,1),YTAB(3,1),
     .YTAB(4,1),P,RADOUT
C
C     FIND NEXT OUTPUT TEMPERATURE
C
  105 ITEMP=ITEMP+1
      IF(ITEMP.GT.40)GOTO 130
      IF(YTAB(1,I).GT.TMPOUT(ITEMP))GOTO 105
C
C     FIND PRINT TEMPERATURE AND PRINT
C
  110 IF(YTAB(1,I+1).GT.TMPOUT(ITEMP))GOTO 115
      I=I+1
      IF(I.GE.I1)GOTO 130
      GOTO 110
  115 FRAC=(TMPOUT(ITEMP)-YTAB(1,I))/(YTAB(1,I+1)-YTAB(1,I))
      YPASS(1)=TMPOUT(ITEMP)
      YPASS(2)=YTAB(2,I)+FRAC*(YTAB(2,I+1)-YTAB(2,I))
      YPASS(3)=YTAB(3,I)+FRAC*(YTAB(3,I+1)-YTAB(3,I))
      YPASS(4)=YTAB(4,I)+FRAC*(YTAB(4,I+1)-YTAB(4,I))
      SPRIN=YTAB(5,I)+FRAC*(YTAB(5,I+1)-YTAB(5,I))
      CALL DIVF(YTAB(5,I),YTAB(1,I),TAB)
      P1=YTAB(2,I)*(1.D0+CHI)*YTAB(1,I)
      CALL DIVF(YTAB(5,I+1),YTAB(1,I+1),TAB)
      P2=YTAB(2,I+1)*(1.D0+CHI)*YTAB(1,I+1)
      P=P1+FRAC*(P2-P1)
      RADOUT=1.D50
      IF(YPASS(2).EQ.0.D0)GOTO 125
      RADOUT=-(P*P)/(OKAP*YPASS(3))
  125 P=KAY*P
      SPRIN=SPRIN-SBOTT
      WRITE(6,6002)SPRIN,TMPOUT(ITEMP),YPASS(2),YPASS(3),
     .YPASS(4),P,RADOUT
      I=I+1
      IF(I.GE.I1)GOTO 130
      ILINE=ILINE+1
      IF(ILINE.LT.58)GOTO 105
      ILINE=0
      WRITE(6,6004)
      GOTO 105
```

```
  130 IF(I.GT.I1)GOTO 99
      SPRIN=YTAB(5,I)
      CALL DIVF(SPRIN,YTAB(1,I),TAB)
      P=YTAB(2,I)*(1.D0+CHI)*YTAB(1,I)
      RADOUT=1.D50
      IF(YTAB(3,I).EQ.0.D0)GOTO 135
      RADOUT=-(P*P)/(OKAP*YTAB(3,I))
  135 P=KAY*P
      SPRIN=SPRIN-SBOTT
      WRITE(6,6002)SPRIN,YTAB(1,I),YTAB(2,I),YTAB(3,I),
     .YTAB(4,I),P,RADOUT
   99 CONTINUE
C
C     WRITE OUT TABULATED RESULTS OF INTEGRATION IF REQUESTED
C
      IF(NOUT.GE.1)CALL WRTR(YTAB,NMAX)
  999 CONTINUE
      STOP
      END
      SUBROUTINE WRTR(Y,N)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 Y(5,N)
 1001 FORMAT(10A8)
 1002 FORMAT(A4)
      WRITE(10,1002)N
      WRITE(10,1001)Y
      RETURN
      END
```

```
      SUBROUTINE ABMINT(S,YINT,F,DS,EPS,TSTOP,SSTOP,MSTOP,YTAB,A0,
     .ITS,PRCT,NMAX)
C
C     ABMINT SOLVES A SET OF THREE COUPLED ORDINARY DIFFERENTIAL
C     EQUATIONS PLUS A CONSERVATION RELATION THAT DESCRIBE THE
C     (STEADY STATE) BEHAVIOR OF A COMPRESSIBLE FLUID IN A FLUX TUBE.
C     THE ROUTINE TAKE THE FOLLOWING INPUT PARAMETERS:
C
C     S  THE INITIAL DISTANCE (ARBITRARY)
C
C     YINT  THE INITIAL VALUES OF Y(1)-Y(4), THE INDEPENDENT
C           VARIABLES (TEMPERATURE, DENSITY, HEAT FLUX AND
C           VELOCITY)
C
C     F  THE NAME OF THE SUBROUTINE THAT CALCULATES THE
C        DERIVATIVES OF THE INDEPENDENT VARIABLE AND THE VELOCITY
C        (MUST BE DECLARED IN AN EXTERNAL STATEMENT IN THE
C        CALLING ROUTINE)
C
C     DS  THE INITIAL STEP SIZE
C
C     EPS  THE DESIRED ACCURACY (RELATIVE) FOR A DISTANCE DS
C
C     TSTOP  THE MAXIMUM (OR MINIMUM) TEMPERATURE TO WHICH THE
C            ROUTINE WILL INTEGRATE
C
C     SSTOP  THE MAXIMUM (MINIMUM) DISTANCE TO WHICH THE ROUTINE
C            WILL INTEGRATE
C
C     MSTOP  THE MAXIMUM MACH NUMBER TO WHICH THE ROUTINE WILL INTEGRATE
C
C     YTAB  AN ARRAY IN WHICH THE RESULTS OF THE INTEGRATION ARE
C           RETURNED TO THE CALLING PROGRAM - SHOULD BE DEMINISIONED AT
C           LEAST 5*NMAX.  VARIABLES STORED IN THE FOLLOWING ORDER:
C           TEMPERATURE, DENSITY,HEAT FLUX,VELOCITY AND DISTANCE
C
C     A0  THE AREA AT THE STARTING POINT TIMES THE DENSITY AT THE
C         STARTING POINT TIMES THE VELOCITY AT THE STARTING POINT
C         (A CONSERVED QUANTITY)
C
C     ITS INDEX OF THE VARIABLE THAT CONTROLS THE FREQUENCY
C         AT WHICH RESULTS ARE PUT IN YTAB
C
C     PRCT  THE MULTIPLICATIVE FACTOR BY WHICH THE ITS ELEMENT OF Y
C           IS ALLOWED TO CHANGE BETWEEN THE TABULATION OF THE RESULTS
C
C     NMAX  THE MAXIMUM NUMBER OF TABULATION POINTS
C
C
C     THE ROUTINE USES SEVERAL LOCAL WORKING ARRAYS
C
C     RUNGE-KUTTA:
C
C           Y(4),Y1(4),F0(4),F1(4),F2(4) USED TO STORE INTERMEDIATE
C           VALUES OF THE INDEPENDENT VARIABLE AND THEIR DERIVATIVES
C
C     ADAMS-BASHFORTH-MOULTON PREDICTOR CORRECTOR:
C
C           YW(32),FW(32) USED TO STORE LAST 8 STEPS OF INTEGRATION.
C           THE PRESENT INTEGRATION USES 4 PREVIOUS VALUES TO ESTIMATE
C           THE NEXT VALUE SO DOUBLING THE STEP SIZE CAN BE DONE IF
C           AT LEAST 4 INTEGRATION STEPS HAVE OCCURRED SINCE THE LAST
C           DOUBLING OF THE STEP SIZE
C
C           YP(4),YC(4)  USED TO STORE THE PREDICTED AND CORRECTED
```

```
C             VALUES OF THE INDEPENDENT VARIABLES
C
C      ABMINT USES AN ADAMS-BASHFORTH-MOULTON PREDICTOR-CORRECTOR
C      INTEGRATION SCHEME.  START UP IS ACCOMPLISHED BY BACKWARD
C      INTEGRATION WITHA RUNGE-KUTTA SCHEME AND MISSING VALUES
C      NEEDED WHEN HALVING THE STEP SIZE ARE PROVIDED USING THE
C      SAME RUNGE-KUTTA SCHEME
C
C
C      DECLARE VARIABLES
C
       IMPLICIT REAL*8 (A-H,O-Z)
       REAL*8 S,EPS,ERR,DS,DT,TSTOP,TDIF1,TDIF2,H2,H3,H6,H8,H,T,ERR1,
      .YP(4),YC(4),YINT(4),YTAB(5,1),FO(4),F1(4),F2(4),Y1(4),Y(4),FP(4),
      .ON24/Z3FAAAAAAAAAAAAB/,ERST,FW(32)/32*0.DO/,YW(32)/32*0.DO/,
      .MSTOP,MACH,SSTOP,SDIF1,SDIF2,H924,CCC1/Z4161C71C71C71C72/,
      .CCC2/Z4168E38E38E38E39/,CCC3/Z4141C71C71D71C72/,FRAC
       INTEGER*4 I,J,K,IND,IN1,IN2,IN3,IN4,IT,DOUBLE
       LOGICAL*4 DONE
C
C      START UP USING INTEGRATION BY 4TH ORDER R-K & 1/32 DS
C
       VCH1=PRCT*YINT(ITS)
       VCH2=PRCT*VCH1
       FRAC=1.DO
       MACH=1.649959D8*MSTOP*MSTOP
       MMAX=NMAX
    15 DT=.03125DO*DS*FRAC
       H2=DT*.5DO
       H3=DT/3.DO
       H6=H3*.5DO
       H8=H2*.25DO
       T=S
       IND=4
       DO 11 I=1,4
    11 YTAB(I,1)=YINT(I)
       YTAB(5,1)=S
       CALL F(T,YINT,FO,AO,&999)
       DO 1 I=1,4
         Y(I)=YINT(I)
         YW(I)=YINT(I)
     1   FW(I)=FO(I)
       DO 2 I=1,3
         DO 3 J=1,2
           DO 4 K=1,3
     4       Y1(K)=FO(K)*H3+Y(K)
           CALL F(T+H3,Y1,F1,AO,&999)
           DO 5 K=1,3
     5       Y1(K)=(FO(K)+F1(K))*H6+Y(K)
           CALL F(T+H3,Y1,F1,AO,&999)
           DO 6 K=1,3
     6       Y1(K)=(F1(K)*3.DO+FO(K))*H8+Y(K)
           CALL F(T+H2,Y1,F2,AO,&999)
           DO 7 K=1,3
     7       Y1(K)=(F2(K)*4.DO-F1(K)*3.DO+FO(K))*H2+Y(K)
           T=T+DT
           CALL F(T,Y1,F1,AO,&999)
           DO 9 K=1,3
     9       Y(K)=(F2(K)*4.DO+F1(K)+FO(K))*H6+Y(K)
     3     CALL F(T,Y,FO,AO,&999)
         DO 8 J=1,4
           YW(IND+J)=Y(J)
     8     FW(IND+J)=FO(J)
     2  IND=IND+4
       IF(YW(12+ITS).LT.VCH1)GOTO 16
       FRAC=FRAC*0.5DO
       GOTO 15
    16 SDIF1=S-SSTOP
       S=T
```

```
      TDIF1=YW(1)-TSTOP
      H=DS*FRAC*0.0625D0*ON24
      DT=DS*FRAC*.0625D0
      H924=DT*.375D0
      ERR=EPS*FRAC*.015625D0
      ERR1=ERR*0.03125D0
      IN1=0
      IN2=4
      IN3=8
      IN4=12
      DOUBLE=4
      DONE=.FALSE.
C
C     END INITIALIZATION
C
      I=2
   50    DOUBLE=DOUBLE-1
   55    CONTINUE
         DO 20 J=1,3
   20      YP(J)=YW(IN4+J)+H924*(CCC1*FW(IN4+J)-CCC2*FW(IN3+J)
     .     +CCC3*FW(IN2+J)-FW(IN1+J))
         CALL F(S+DT,YP,FP,A0,&999)
         DO 30 J=1,3
   30      YC(J)=YW(IN4+J)+H*(9.*FP(J)+19.*FW(IN4+J)-5.*FW(IN3+J)
     .     +FW(IN2+J))
         TDIF2=YC(1)-TSTOP
         IF(DSIGN(TDIF2,TDIF1).NE.TDIF2)DONE=.TRUE.
         SDIF2=S-SSTOP
         IF(DSIGN(SDIF2,SDIF1).NE.SDIF2)DONE=.TRUE.
   35    CALL F(S+DT,YC,FP,A0,&999)
         IF(YC(4)*YC(4).GT.MACH*YC(1))DONE=.TRUE.
         ERST=DABS(YP(1)-YC(1))/(DABS(YP(1))+DABS(YC(1)))
         ERST=DMAX1(DABS(YP(2)-YC(2))/(DABS(YP(2))+DABS(YC(2))),ERST)
         IF(DABS(YC(3))+DABS(YP(3)).LT.1.E-30)GOTO 220
         ERST=DMAX1(DABS(YP(3)-YC(3))/(DABS(YP(3))+DABS(YC(3))),ERST)
  220    IF(DABS(YC(4))+DABS(YP(4)).LT.1.E-30)GOTO 215
         ERST=DMAX1(DABS(YP(4)-YC(4))/(DABS(YP(4))+DABS(YC(4))),ERST)
  215    IF(YC(ITS).GT.VCH2)GOTO 90
         IF(ERST.GT.ERR)GOTO 100
         IF(ERST.LT.ERR1)GOTO 200
  205    S=S+DT
         IN1=IN2
         IN2=IN3
         IN3=IN4
         IN4=MOD(IN4+4,32)
         DO 40 J=1,4
           YW(IN4+J)=YC(J)
   40      FW(IN4+J)=FP(J)
         IF(YC(ITS).LT.VCH1)GOTO 50
        DO 60 J=1,4
   60    YTAB(J,I)=YC(J)
      YTAB(5,I)=S
      I=I+1
      IF(I.GT.NMAX)GOTO 10
      VCH1=VCH2
      VCH2=PRCT*VCH1
      IF(DONE)GOTO 10
      GOTO 50
   90   IACC=1
        GOTO 103
  100   IACC=0
  103   CONTINUE
C
C     SECTION THAT HALVES INTEGRATION STEP
C
        IF(DT.LT.1.D-1)GOTO 205
        IF(DT.GT.DS)GOTO 109
        ERR=ERR*.5D0
        ERR1=ERR1*.5D0
```

```
109   DT=DT*.5D0
      H=DT*ON24
      H924=DT*.375D0
C
C    REARRANGE WORKING ARRAYS
C
      IT=MOD(IN4+12,32)
      J=IN4
      DO 101 K=1,4
        FW(IT+K)=FW(IN4+K)
101     YW(IT+K)=YW(IN4+K)
      IN4=IT
      IT=MOD(IN3+8,32)
      DO 102 K=1,4
        FW(IT+K)=FW(IN3+K)
        YW(IT+K)=YW(IN3+K)
        FW(IN3+K)=FW(IN2+K)
102     YW(IN3+K)=YW(IN2+K)
      IN2=IT
      IT=IN3
      IN1=J
      IN3=MOD(IN2+4,32)
C
C    GENERATE MISSING INFORMATION WITH 4TH ORDER R-K
C
      H2=DT*.25D0
      H3=DT/6.D0
      H6=H3*.5D0
      H8=H2*.25D0
      T=S-(DT+DT)
      DO 110 J=1,4
        Y(J)=YW(IN2+J)
110     FO(J)=FW(IN2+J)
      DO 120 J=1,2
        DO 130 K=1,3
130       Y1(K)=FO(K)*H3+Y(K)
        CALL F(T+H3,Y1,F1,AO,&999)
        DO 140 K=1,3
140       Y1(K)=(FO(K)+F1(K))*H6+Y(K)
        CALL F(T+H3,Y1,F1,AO,&999)
        DO 150 K=1,3
150       Y1(K)=(F1(K)*3.DO+FO(K))*H8+Y(K)
        CALL F(T+H2,Y1,F2,AO,&999)
        DO 160 K=1,3
160       Y1(K)=(F2(K)*4.DO-F1(K)*3.DO+FO(K))*H2+Y(K)
        T=T+H2+H2
        CALL F(T,Y1,F1,AO,&999)
        DO 180 K=1,3
180       Y(K)=(F2(K)*4.DO+F1(K)+FO(K))*H6+Y(K)
        CALL F(T,Y,FO,AO,&999)
120     CONTINUE
      DO 170 J=1,4
        FW(IN3+J)=FO(J)
170     YW(IN3+J)=Y(J)
      DO 115 J=1,4
        Y(J)=YW(IT+J)
115     FO(J)=FW(IT+J)
      T=S-4.DO*DT
      DO 125 J=1,2
        DO 135 K=1,3
135       Y1(K)=FO(K)*H3+Y(K)
        CALL F(T+H3,Y1,F1,AO,&999)
        DO 145 K=1,3
145       Y1(K)=(FO(K)+F1(K))*H6+Y(K)
        CALL F(T+H3,Y1,F1,AO,&999)
        DO 155 K=1,3
155       Y1(K)=(F1(K)*3.DO+FO(K))*H8+Y(K)
        CALL F(T+H2,Y1,F2,AO,&999)
        DO 165 K=1,3
```

```
 165        Y1(K)=(F2(K)*4.D0-F1(K)*3.D0+F0(K))*H2+Y(K)
          T=T+H2+H2
          CALL F(T,Y1,F1,A0,&999)
          DO 185 K=1,3
 185        Y(K)=(F2(K)*4.D0+F1(K)+F0(K))*H6+Y(K)
          CALL F(T,Y,F0,A0,&999)
 125      CONTINUE
        DO 175 J=1,4
          FW(IN1+J)=F0(J)
 175      YW(IN1+J)=Y(J)
        GOTO 55
C
C     RETURN TO ABM P-C INTEGRATION WITH NEW STEP SIZE
C
 200    CONTINUE
C
C     SECTION THAT DOUBLES INTEGRATION STEP SIZE
C
        IF(IACC.EQ.1)GOTO 205
        IF(DOUBLE.GE.0)GOTO 205
        DOUBLE=4
        S=S-DT
        DT=DT+DT
        H=DT*ON24
        H924=DT*.375D0
        IF(DT.GT.DS)GOTO 209
        ERR=ERR+ERR
        ERR1=ERR1+ERR1
 209    K=MOD(IN4+4,32)
        IT=MOD(IN4+12,32)
        DO 210 J=1,4
          FW(IN4+J)=FW(IN3+J)
          YW(IN4+J)=YW(IN3+J)
          FW(IN3+J)=FW(IN1+J)
          YW(IN3+J)=YW(IN1+J)
          FW(IN1+J)=FW(K+J)
          YW(IN1+J)=YW(K+J)
          FW(IN2+J)=FW(IT+J)
 210      YW(IN2+J)=YW(IT+J)
        GOTO 205
  10  NMAX=I-1
      RETURN
 999  CONTINUE
6001  FORMAT(1H ,'FATAL ERROR T OR S OUT OF TABULATED RANGE')
      WRITE(6,6001)
      S=-S
      NMAX=I
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY



```
DIVF     CSECT
*
*        DIVF(S,Y,DY,AO,*)          OR    DIVF(S,Y,TAB)
*        REAL*8 S,Y(4),DY(4),AO           REAL*8 S,Y(4),TAB(8)
*
*        THE FIRST FORM OF THE CALL CALCUALTES THE DERIVATIVES
*
*        DT/DS  = DY(1)/DS , DN/DS = DY(2)/DS & DQ/DS = DY(3)/DS
*
*        AND STORES THEN IN ARRAY DY.  THE VALUE OF  V=Y(4) IS COMPUTED
*        FROM THE CONSERVATION LAW NVA = CONSTANT AND STORED IN Y(4).
*        THE ROUTINE INTERPOLATES THE VALUES OF PRARMETERS NEEDED FOR
*        THE CALCULATIONS FROM TABULATIONS OF 4 FUNCTIONS OF S ONLY AND
*        4 FUNCTIONS OF T ONLY.  IF S OR T IS OUT OF THE TABULATED
*        RANGE, THE OFFENDING QUANTITY IS NEGATED AND THE ROUTINE DOES
*        THE EQUIVALENT OF A FORTRAN RETURN1.
*
*        THE SECOND FORM OF THE CALL STATEMENT (DISTINGUISHED FROM THE
*        FIRST BY THE NUMBER OF ARGUMENTS) CALCULATES THE INTERPOLATED
*        VALUES OF G(S),DA/DS(S),SOURCE(S), AREA(S) AND 1/KAPPA(T),
*        LAMBDA(T), CHI(T) AND DCHI/DT(T) AND STORES THEM IN TAB.
*
*        THERE IS A SECOND ENTRY POINT (DIVINT) WHICH PICKS UP
*        AND STORES LOCALLY THE ADDRESSES OF THE TABULATIONS OF
*        THE FUNCTIONS NEEDED FOR THE CALCULATIONS.
*
*        NOTE THAT THIS MEANS THAT MEANINGLESS RESULTS
*        WILL BE PRODUCED IF DIVINT IS NOT CALLED BEFORE
*        THE FIRST TIME DIVF IS CALLED.  IT IS EVEN POSSIBLE
*        THAT SOME SORT OF ABEND WILL RESULT.
*
*
*  THE FOLLOWING TWO FORTRAN SUBROUTINES ARE ROUGHLY EQUIVALENT
*  TO THE TWO CALLS TO DIVF (DIVINT IS NOT REPRODUCED)
*
*     SUBROUTINE DIVF(S,Y,DY,AO,*)
*     IMPLICIT REAL*8 (A-H,O-Z)
*     DIMENSION Y(4),DY(4)
*     REAL*8 KOMH/8.2989776D7/,TION/1.046460D5/,KAY/1.38062D-16/
*     DY(1)=-Y(3)*OKAP(T)
*     Y(4)=AO/(Y(2)*A(S))
*     DY(2)=(Y(2)*(Y(4)*Y(4)*DADS(S)-DY(1)*KOMH
*    .((1.DO+CHI(T))+(T*DCHI(T))-G(S)*S)))/
*    .(KOMH*(1.DO+CHI(T))*Y(1)-Y(4)*Y(4))
*     DY(3)=-(L(T)*CHI(T)*Y(2)*Y(2))-(1.5DO*Y(4)*Y(2)*KAY*DY(1)
*    .*((1.DO+CHI(T))+(Y(1)-TION)*DCHI(T)))+((1.DO_CHI(T))*KAY
*    .*Y(1)*Y(4)*DY(2))+SOR(S)-Y(3)*DADS
*     RETURN
*     END
*
*     SUBROUTINE DIVF(S,Y,TAB)
*     IMPLICIT REAL*8 (A-H,O-Z)
*     DIMENSION Y(4),TAB(8)
*     TAB(1) G(S)
*     TAB(2)=DADS(S)
*     TAB(3)=SOR(S)
*     TAB(4)=A(S)
*     TAB(5)=OKAP(T)
*     TAB(6)=L(T)
*     TAB(7)=CHI(T)
*     TAB(8)=DCHI(T)
*     RETURN
*     END
*
*        NOTE:  IN THE COMMENTS 'R' REFERS TO GENERAL PURPOSE REGISTERS
```

```
*        AND 'F' REFERS TO FLOATING POINT REGISTERS.
*
         USING *,15           TELL ASSEMBLER NEST INST ADDR IN R15
         B     DFIRST         BRANCH AROUND NAME AND OTHER ENTRY POINT
         DC    X'04'
         DC    CL5'DIVF '
*
*        DIVINT(G,DADS,SOR,AREA,OKAP,LUM,CHI,DCHI)
*        REAL*8 G(563),DADS(563),SOR(563),AREA(563),
*        REAL*8 OKAP(820),LUM(820),CHI(820),DCHI(820)
*
         ENTRY DIVINT
         USING *,15           TELL ASSEMBLER NEXT INSTR ADDR IN R15
DIVINT   B     TFIRST         BRANCH AROUND NAME
         DC    X'06'
         DC    CL7'DIVINT '
TFIRST   STM   14,12,12(13)   SAVE CALLING ROUTINES GPR'S
         LM    2,9,0(1)       GET BASE ADDR'S OF INTERPOLATION TABLES
         STM   2,9,GADDR      SAV TABLE BASE ADDR'S
         LM    2,9,28(13)     RESTORE CALLING ROUTINE'S GPR'S
         MVI   12(13),X'FF'   INDICATE CONTROL RETURNED
         BR    14             RETURN FROM INITIALIZATION
*
*        MAIN ROUTINE RESUMES
*
         USING DIVF,15
DFIRST   STM   14,12,12(13)   SAVE CALLING ROUTINES GPR'S
         L     2,0(1)         R2 <= ADDR S
         LM    3,6,GADDR      R3-R6 <= ADDR'S OF TABLES FOR S
         MVC   FLOAT+1(6),2(2) FLOAT <= FRACTIONAL DISTANCE FROM
*                                NEXT SMALLER VALUE OF S TABULATED.
         LH    12,0(2)        R12 <= HIGH ORDER BYTES OF S
         S     12,SDISP       REDUCE R12 BY SDISP - # WORDS FROM
*                             BASE OF TABLES
         LD    4,FLOAT        F4 <= FRACTION 0 LE FRAC LE 1
         BM    BADS           IF RESULT IS NEGATIVE OUT OF RANGE
*                             GOTO BADS
         C     12,SBND        COMPARE R12 TO SBND - IF GREATER
         BH    BADS           OUT OF RANGE GOTO BADS
         SD    4,=D'.5'       F4 <= X = FRAC - .5  -.5 LE X LE .5
         SLA   12,3           R12 <= R12*8 NOW BYTE DISPLACEMENT
*                             FROM BASE OF INTERPOLATION TABLE.
*
*        NOW COMPUTE WEIGHTS FOR CUBIC INTERPOALTION OF
*        FUNCTIONS OF S
*
         LDR   2,4            F2 <= X
         MDR   4,4            F4 <= X**2 = X2
         HDR   4,4            F4 <= X2/2
         SD    4,=D'1.125'    F4 <= X2/2 - 9/8
         LDR   6,4            F6 <= X2/2 - 9/8
         HDR   4,4            F4 <= X2/4 - 9/16
         MDR   6,2            F6 <= X3/2 - 9X/8
         LCDR  0,4            F0 <= -X2/4 + 9/16
         ADR   0,6            F0 <= X3/2 - X2/4 - 9X/8 + 9/16
         STD   0,WM1          WM1 <= WEIGHT FOR TABLE ENTRY CORRESPOND-
*                             ING TO CLOSEST SMALLER VALUE OF S.
         LCDR  0,4            F0 <= -X2/4 + 9/16
         SDR   0,6            F0 <= -X3/2 - X2/4 + 9X/8 + 9/16
         STD   0,WP1          WP1 <= WEIGHT FOR TABLE ENTRY CORRESPOND-
*                             ING TO CLOSEST LARGER VALUE OF S.
         ADR   6,2            F6 <= X3/2 - X/8
         AD    4,=D'.5'       F4 <= X2/4 - 1/16
         MD    6,=X'4055555555555555'  F6 <= X3/6 - X/24
         LDR   0,4            F0 <= X2/4 - 1/16
         SDR   0,6            F0 <= -X3/6 + X2/4 + X/24 - 1/16
         ADR   6,4            F6 <= X3/6 + X2/4 - X/24 - 1/16
         STD   0,WM3          WM3 <= WEIGHT FOR TABLE ENTRY CORRESPOND-
*                             ING TO 2ND CLOSEST SMALLER VALUE OF S.
```

```
         STD    6,WP3              WP3 <= WEIGHT FOR TABLE ENTRY CORRESPOND-
*                                  ING TO 2ND CLOSEST LARGER VALUE OF S.
*
* NOW CALCULATE INTERPOLATED VALUES OF GRAVITY AND DA/DS
* (HAVE WEIGHTS FOR TABLES ENTRIES 1 & 4 IN FPR'S 0 & 6)
*
         LDR    4,0                F4 <= WEIGHT 1
         LDR    2,6                F2 <= WEIGHT 4
         MD     0,0(3,12)          F0 <= WEIGHT 1 * GRAV 1
         MD     2,24(3,12)         F2 <= WEIGHT 4 * GRAV 4
         MD     4,0(4,12)          F4 <= WEIGHT 1 * DA/DS 1
         MD     6,24(4,12)         F6 <= WEIGHT 4 * DA/DS 4
         ADR    0,2                F0 <= W1*G1 + W4*G4
         ADR    4,6                F4 <= W1*D1 + W4*D4
         LD     2,WM1              F2 <= WEIGHT 2
         LDR    6,2                F6 <= WEIGHT 2
         MD     2,8(3,12)          F2 <= W2*G2
         MD     6,8(4,12)          F6 <= W2*D2
         ADR    0,2                F0 <= W1*G1 + W4*G4 + W2*G2
         ADR    4,6                F4 <= W1*D1 + W4*D4 + W2*D2
         LD     2,WP1              F2 <= WEIGHT 3
         LDR    6,2                F6 <= WEIGHT 3
         MD     2,16(3,12)          F2 <= W3*G3
         MD     6,16(4,12)          F6 <= W3*D3
         ADR    0,2                F2 <= INTERPOLATED VALUE OF G
         ADR    4,6                F4 <= INTERPOLATED VALUE OF DA/DS
         STD    0,G                G <= INTERPOLATED VALUE OF GRAVITY
         STD    4,DADS             DADS <= INTERPOLATED VALUE OF DA/DS
         LD     0,WM3              F0 <= WEIGHT 1
         LD     2,WP3              F2 <= WEIGHT 4
*
* NOW CALCULATE  VALUES OF SOURCE AND AREA
* ( HAVE WEIGHTS 1 & 4 IN FPR'S 0 & 2)
*
         LDR    4,0                F4 <= WEIGHT 1
         LDR    6,2                F6 <= WEIGHT 4
         MD     0,0(5,12)          F0 <= WEIGHT 1 * SOURCE 1
         MD     2,24(5,12)         F2 <= WEIGHT 4 * SOURCE 4
         MD     4,0(6,12)          F4 <= WEIGHT 1 * AREA 1
         MD     6,24(6,12)         F6 <= WEIGHT 4 * AREA 4
         ADR    0,2                F0 <= W1*S1 + W4*S4
         ADR    4,6                F4 <= W1*A1 + W4*A4
         LD     2,WM1              F2 <= WEIGHT 2
         LDR    6,2                F6 <= WEIGHT 2
         MD     2,8(5,12)          F2 <= W2*S2
         MD     6,8(6,12)          F6 <= W2*A2
         ADR    0,2                F0 <= W1*S1 + W4*S4 + W2*S2
         ADR    4,6                F4 <= W1*A1 + W4*A4 + W2*A2
         LD     2,WP1              F2 <= WEIGHT 3
         LDR    6,2                F6 <= WEIGHT 3
         MD     2,16(5,12)          F2 <= W3*S3
         MD     6,16(6,12)          F6 <= W3*A3
         ADR    0,2                F2 <= INTERPOLATED VALUE OF SOURCE
         ADR    4,6                F4 <= INTERPOLATED VALUE OF AREA
         STD    0,SOR              SOR <= INTERPOLATED VALUE OF SOURCE
         STD    4,AREA             AREA <= INTERPOLATED VALUE OF AREA
*
* CALCULATE INDEX AND FRACTIONAL DISPLACEMENT FOR INTERPOLATION
* ON TEMPERATRUE (T) TABLE
*
         L      2,4(1)             R2 <= BASE ADDR Y ARRAY
         LM     3,6,KADDR          R3-R6 <= BASE ADDR'S TABLES FOR
*                                  INTERPOLATION OF FUNCTIONS OF T
         MVC    FLOAT+1(6),2(2) FLOAT <= FRACTIONAL DISTANCE FROM
*                                     NEXT SMALLER VALUE OF T TABULATED.
         LH     12,0(2)            R12 <= HIGH ORDER BYTES OF T
         S      12,TDISP           REDUCE R12 BY TDISP - # WORDS FROM
*                                  BASE OF TABLES
         LD     4,FLOAT            F4 <= FRACTION 0 LE FRAC LE 1
```

```
         BM     BADS             IF RESULT IS NEGATIVE OUT OF RANGE
*                                GOTO BADS
         C      12,TBND          COMPARE R12 TO TBND - IF GREATER
         BH     BADS             OUT OF RANGE GOTO BADS
         SD     4,=D'.5'         F4 <= X = FRAC - .5   -.5 LE X LE .5
         SLA    12,3             R12 <= R12*8 NOW BYTE DISPLACEMENT
*                                FROM BASE OF INTERPOLATION TABLE.
*
*        NOW COMPUTE WEIGHTS FOR CUBIC INTERPOALTION OF
*        FUNCTIONS OF T
*
         LDR    2,4              F2 <= X
         MDR    4,4              F4 <= X**2 = X2
         HDR    4,4              F4 <= X2/2
         SD     4,=D'1.125'      F4 <= X2/2 - 9/8
         LDR    6,4              F6 <= X2/2 - 9/8
         HDR    4,4              F4 <= X2/4 - 9/16
         MDR    6,2              F6 <= X3/2 - 9X/8
         LCDR   0,4              F0 <= -X2/4 + 9/16
         ADR    0,6              F0 <= X3/2 - X2/4 - 9X/8 + 9/16
         STD    0,WM1            WM1 <= WEIGHT FOR TABLE ENTRY CORRESPOND-
*                                ING TO CLOSEST SMALLER VALUE OF T.
         LCDR   0,4              F0 <= -X2/4 + 9/16
         SDR    0,6              F0 <= -X3/2 - X2/4 + 9X/8 + 9/16
         STD    0,WP1            WP1 <= WEIGHT FOR TABLE ENTRY CORRESPOND-
*                                ING TO CLOSEST LARGER VALUE OF T.
         ADR    6,2              F6 <= X3/2 - X/8
         AD     4,=D'.5'         F4 <= X2/4 - 1/16
         MD     6,=X'4055555555555555'  F6 <= X3/6 - X/24
         LDR    0,4              F0 <= X2/4 - 1/16
         SDR    0,6              F0 <= -X3/6 + X2/4 + X/24 - 1/16
         ADR    6,4              F6 <= X3/6 + X2/4 - X/24 - 1/16
         STD    0,WM3            WM3 <= WEIGHT FOR TABLE ENTRY CORRESPOND-
*                                ING TO 2ND CLOSEST SMALLER VALUE OF T.
         STD    6,WP3            WP3 <= WEIGHT FOR TABLE ENTRY CORRESPOND-
*                                ING TO 2ND CLOSEST LARGER VALUE OF T.
*
* NOW CALCULATE INTERPOLATED VALUES OF 1/KAPPA(T) AND L
* (HAVE WEIGHTS FOR TABLES ENTRIES 1 & 4 IN FPR'S 0 & 6)
*
         LDR    4,0               F4 <= WEIGHT 1
         LDR    2,6               F2 <= WEIGHT 4
         MD     0,0(3,12)         F0 <= WEIGHT 1 * K 1
         MD     2,24(3,12)        F2 <= WEIGHT 4 * K 4
         MD     4,0(4,12)         F4 <= WEIGHT 1 * L 1
         MD     6,24(4,12)        F6 <= WEIGHT 4 * L 4
         ADR    0,2               F0 <= W1*K1 + W4*K4
         ADR    4,6               F4 <= W1*L1 + W4*L4
         LD     2,WM1             F2 <= WEIGHT 2
         LDR    6,2               F6 <= WEIGHT 2
         MD     2,8(3,12)         F2 <= W2*K2
         MD     6,8(4,12)         F6 <= W2*L2
         ADR    0,2               F0 <= W1*K1 + W4*K4 + W2*K2
         ADR    4,6               F4 <= W1*L1 + W4*L4 + W2*L2
         LD     2,WP1             F2 <= WEIGHT 3
         LDR    6,2               F6 <= WEIGHT 3
         MD     2,16(3,12)         F2 <= W3*K3
         MD     6,16(4,12)         F6 <= W3*L3
         ADR    0,2               F2 <= INTERPOLATED VALUE OF 1/KAPPA
         ADR    4,6               F4 <= INTERPOLATED VALUE OF L
         STD    0,OKAP            OKAP <= INTERPOLATED VALUE OF 1/KAPPA
         STD    4,LUM             LUM <= INTERPOLATED VALUE OF L
*
* CHECK TO SEE IF GAS FULLY IONIZED (T > 65,536, BYTE DISP > 3BC(HEX)),
* IF SO SKIP INTERPOLATION OF CHI AND DCHI/DT, IF NOT CONTINUE
*
         C      12,HBYTE
         BC     2,HIGHT
         LD     0,WM3             F0 <= WEIGHT 1
```

```
         LD     2,WP3             F2 <= WEIGHT 4
*
* NOW CALCULATE  VALUES OF CHI AND DCHI/DT
* ( HAVE WEIGHTS 1 & 4 IN FPR'S 0 & 2)
*
         LDR    4,0               F4 <= WEIGHT 1
         LDR    6,2               F6 <= WEIGHT 4
         MD     0,0(5,12)         F0 <= WEIGHT 1 * CHI 1
         MD     2,24(5,12)        F2 <= WEIGHT 4 * CHI 4
         MD     4,0(6,12)         F4 <= WEIGHT 1 * DCHI/DT 1
         MD     6,24(6,12)        F6 <= WEIGHT 4 * DCHI/DT 4
         ADR    0,2               F0 <= W1*C1 + W4*C1
         ADR    4,6               F4 <= W1*D1 + W4*D4
         LD     2,WM1             F2 <= WEIGHT 2
         LDR    6,2               F6 <= WEIGHT 2
         MD     2,8(5,12)         F2 <= W2*C2
         MD     6,8(6,12)         F6 <= W2*D2
         ADR    0,2               F0 <= W1*C1 + W4*C4 + W2*C2
         ADR    4,6               F4 <= W1*D1 + W4*D4 + W2*D2
         LD     2,WP1             F2 <= WEIGHT 3
         LDR    6,2               F6 <= WEIGHT 3
         MD     2,16(5,12)         F2 <= W3*C3
         MD     6,16(6,12)         F6 <= W3*D3
         ADR    0,2               F2 <= INTERPOLATED VALUE OF CHI
         ADR    4,6               F4 <= INTERPOLATED VALUE OF DCHI/DT
         STD    0,CHI             CHI <= INTERPOLATED VALUE OF CHI
         STD    4,DCHI            DCHI <= INTERPOLATED VALUE OF DCHI/DT
         LM     3,4,8(1)           GPR'S 3&4 <= BASE ADDR'S OF DY AND AO
         LTR    3,3              CHECK IF 3 NEGATIVE IF SO IS LAST
         BM     TABL             PARAMETER   -  GOTO TABL
         LD     2,OKAP            F2 <= 1/KAPPA
         LNDR   2,2               F2 <= -1/KAPPA
         MD     2,16(2,0)          F2 <= -Q/KAPPA = DT/DS
         STD    2,0(3,0)          STORE DT/DS
         AD     0,=D'1.'          F0 <= 1+CHI
         LD     2,0(4,0)          F2 <= AO = NO*UO*AO
         LD     6,AREA            F6 <= AREA
         MD     6,8(2,0)          F6 <= AREA*N
         DDR    2,6               F2 <= (NO*UO*AO)/(AREA*N) = U
         STD    2,24(2,0)         STORE U (VELOCITY)
         MDR    2,2               F2 <= U**2
         LDR    6,0               F6 <= 1+CHI
         MD     6,KOMH            F6 <= (K*(1+CHI))/MH
         MD     6,0(0,2)          F6 <= (KT*(1+CHI))/MH
         SDR    6,2               F6 <= (KT*(1+CHI))/MH - U**2
         MD     2,DADS            F2 <= DADS*U**2
         SD     2,G               F2 <= DADS*U**2 - GS
         MD     4,0(2,0)          F4 <= DCHI/DT*T
         ADR    4,0               F4 <= (1+CHI) + DCHI/DT*T
         MD     4,KOMH            F4 <= K/MH*F4(¢)
         MD     4,0(3,0)          F4 <= DT/DS*F4(¢)
         SDR    2,4               F2 <= F2(¢) - F4(¢)
         MD     2,8(2,0)          F2 <= F2(¢)*N:
*
* F2 <= N*(U**2*DA/DS - DT/DS*K/MH*((1+CHI)+T*DCHI/DT) - GS)
*
         DDR    2,6               F2 <= DN/DS
         STD    2,8(3,0)          STORE DN/DS
         MDR    2,0               F2 <= (1+CHI)*DN/DS
         LD     4,0(2,0)          F4 <= T
         MDR    2,4               F2 <= T*(1+CHI)*DN/DS
         SD     4,TION            F4 <= T - TI
         MD     4,DCHI            F4 <= (T-TI)*DCHI/DT
         ADR    4,0               F4 <= (1+CHI) + (T-TI)*DCHI/DT
         LD     0,8(2,0)          F0 <= N
         MDR    4,0               F4 <= N*( 1+CHI +(T-TI)*DCHI/DT)
         MD     4,=D'1.5'         F4 <= 3/2 * F4(¢)
         MD     4,0(3,0)          F4 <= DT/DS * F4(¢):
*
```

```
* F4 <= 3/2*DT/DS*(1+CHI + (T-TI)*DCHI/DT)
*
         SDR    2,4              F2 <= (1+CHI)*DN/DS*T - F4(¢)
         MD     2,24(2,0)        F2 <= F2(6) * U
         MD     2,KAY            F2 <= K * F2(¢)
         MDR    0,0              F0 <= N**2
         MD     0,CHI            F0 <= N**2 * CHI
         MD     0,LUM            F0 <= RADIATIVE LOSSES
         SDR    2,0              F2 <= F2(¢) - F0(¢)
         LD     0,16(2,0)         F0 <= Q
         MD     0,DADS           F0 <= Q/A*DA/DS
         SDR    2,0              F2 <= DQ/DS LESS SOURCE TERM
         AD     2,SOR            F2 <= DQ/DS
         STD    2,16(3,0)         STORE DQ/DS
         LM     14,12,12(13)     GPR'S RETURNED TO ORIGINAL STATE
         MVI    12(13),X'FF'     TELL CALLING PROGRAM WE'RE RETURNING
         BCR    15,14            RETURN
*
* END OF SECTION FOR T , 65,536 NEXT SECTION DOES SAME CALCULATIONS
* FOR FULLY IONIZED CASE
*
HIGHT    LM     3,4,8(1)         FPR'S 3&4 <= BASE ADDR'S OF DY & A0
         MVC    0KAP+16(16),=X'41100000000000000000000000000000'
*                               CHI <= 1 & DCHI <= 0
         LTR    3,3             CHECK IF 3 NEGATIVE IF SO IS LAST
         BM     TABL            PARAMETER  -  GOTO TABL
         MD     0,16(2,0)         F0 <= Q/KAPPA
         LCDR   0,0              F0 <= -Q/KAPPA = DT/DS
         STD    0,0(3,0)         STORE DT/DS
         LD     2,0(4,0)         F2 <= A0 = N0*U0*A0
         LD     4,AREA           F4 <= AREA
         MD     4,8(2,0)         F4 <= AREA*N
         DDR    2,4              F2 <= (N0*U0*A0)/(AREA*N) = U
         STD    2,24(2,0)        STORE U (VELOCITY)
         MDR    2,2              F2 <= U**2
         LDR    6,0              F6 <= DT/DS
         LD     4,KOMH2          F4 <= 2K/MH
         MDR    6,4              F6 <= 2K/MH*DT/DS
         MD     4,0(2,0)         F4 <= 2KT/MH
         SDR    4,2              F4 <= 2KT/MH - U**2
         MD     2,DADS           FS <= U**2/A*DA/DS
         SDR    2,6              F2 <= U**2/A*DA/DS - 2K/MH*DT/DS
         SD     2,G              F2 <= F2(¢) - G
         LD     6,8(2,0)         F6 <= N
         MDR    2,6              F2 <= F2(¢) * N
         DDR    2,4              F2 <= DN/DS
         STD    2,8(3,0)         STORE DN/DS
         MDR    0,6              F0 <=N*DT/DS
         LDR    4,0              THIS AND THE NEXT TWO INSTRUCTIONS
         ADR    0,0              EFFECTIVELY MULTIPLY F0 BY 3.
         ADR    0,4              F0 <= 3N*DT/DS
         MD     2,0(2,0)         F2 <= DN/DS*T
         ADR    2,2              F2 <= 2T*DN/DS
         SDR    2,0              F2 <= 2T*DN/DS - 3N*DT/DS
         MD     2,KAY            F2 <= K*(2T*DN/DS - 3N*DT/DS)
         MD     2,24(2,0)        F2 <= K*U*(2T*DN/DS - 3N*DT/DS)
         MDR    6,6              F6 <= N**2
         MD     6,LUM            F6 <= RADIATIVE LOSSES
         SDR    2,6              F2 <= F2(¢) - F6(¢)
         LD     4,16(2,0)         F4 <= Q
         MD     4,DADS           F4 <= Q/A*DA/DS
         SDR    2,4              F2 <= DQ/DS LESS SOURCE TERM
         AD     2,SOR            F2 <= DQ/DS
         STD    2,16(3,0)         STORE DQ/DS
         LM     14,12,12(13)     GPR'S RETURNED TO ORIGINAL STATE
         MVI    12(13),X'FF'     TELL CALLING PROGRAM WE'RE RETURNING
         BR     14               RETURN
*
*        RETURN INTERPOLATED FUNCTION VALUES  -  NOT DERIVITIVES
```

```
*
TABL     MVC   0(64,3),G       PUT INTERPOLATED VALUES IN TAB
         LM    14,12,12(13)    RESTORE GPR'S
         MVI   12(13),X'FF'    INDICATE CONTROL RETURNED
         BR    14              RETURN
*
* CASE OF S OR T OUTSIDE OF THE TABULATED RANGE
*
BADS     LD    0,0(2,0)         F0 <= OFFENDING QUANTITY (T OR S)
         LNDR  0,0              F0 NOW NEGATIVE
         STD   0,0(2,0)         STORE OFFENDING QUANTITY - FLAG
         LM    14,12,12(13)     GPR'S RETURNED TO ORIGINAL STATE
         LA    15,4             GPR 14 <= 4 (RETURN 1)
         MVI   12(13),X'FF'     TELL CALLING PROGRAM WE'RE RETURNING
         BR    14
         CNOP  4,8
*
* STORAGE FOR ADDR'S, CONSTANTS, AND INTERNAL VARIABLES
*
GADDR    DC    X'00000000'
DADDR    DC    X'00000000'
SADDR    DC    X'00000000'
AADDR    DC    X'00000000'
KADDR    DC    X'00000000'
LADDR    DC    X'00000000'
CADDR    DC    X'00000000'
BCDDR    DC    X'00000000'
SDISP    DC    X'00004710'
TDISP    DC    X'00004410'
TBND     DC    X'00000330'
SBND     DC    X'00000330'
HBYTE    DC    X'00000778'
WM3      DC    X'0000000000000000'
WM1      DC    X'0000000000000000'
WP1      DC    X'0000000000000000'
WP3      DC    X'0000000000000000'
G        DC    X'0000000000000000'
DADS     DC    X'0000000000000000'
SOR      DC    X'0000000000000000'
AREA     DC    X'0000000000000000'
OKAP     DC    X'0000000000000000'
LUM      DC    X'0000000000000000'
CHI      DC    X'0000000000000000'
DCHI     DC    X'0000000000000000'
TION     DC    X'45198C6100000000'
KAY      DC    X'339F2CB600000000'
KOMH     DC    X'474EAB1B00000000'
KOMH2    DC    X'479D5A3600000000'
FLOAT    DC    X'4000000000000000'
         END
```

ORIGINAL PAGE IS
OF POOR QUALITY

## REFERENCES


Acton, F. S. 1970 Numerical Methods that Work, (New York: Harper and Row).

Adams, W. M. and Sturrock, P. A. 1975 Ap. J. 202, 259.

Alfven, H. 1938 Phys. Rev. 54, 97.

Alfven, H. 1939 Phys. Rev. 55, 425.

Alfven, H. and Carlqvist, P. 1967 Solar Phys. 1, 220.

Anderson, K. A. and Winckler, J. R. 1962 J. Geophys. Res. 67, 4103.

Appleton, E. V. 1945 Nature 156, 534.

Appleton, E. V. and Hey, J. S. 1946 Phil. Mag. 37, 73.

Barnes, C. W. and Sturrock, P. A. 1972 Ap. J. 174, 654.

Bennett, W. H. 1934 Phys. Rev. 45, 890.

Bray, R. J. and Loughhead, R. E. 1964 Sunspots (London: Chapman and Hall).

Brown, J. C. 1971 Solar Phys. 24, 414.

Brown, J. C. 1972 Solar Phys. 25, 158.

Brown, J. C. 1974 Coronal Disturbances, G. Newkirk, Jr., ed. (Dordrecht: Reidel), p. 395.

Brown, J. C. 1975 Solar Gamma-, X- and EUV Radiation, S. R. Kane, ed. (Dordrecht: Reidel), p. 245.

Brown, J. C. 1976, Phil. Trans. Roy. Soc. Lond. A281, 473.

Brown, J. C. and Melrose, D. B. 1977 Solar Phys. 52, 117.

Brueckner, G. E. 1976 Phil. Trans. Roy. Soc. Lond. A281, 443.

Buneman, O. 1958 Phys. Rev. Lett. 1, 8.

Burnight, T. R. 1949 Phys. Rev. 76, 165.

Carrington, R. C. 1859 M. N. R. A. S. 20, 13.

Chu, K. R. and Rostoker, N. 1973 Phys. Fluids 16, 1472.

Chubb, T. A. 1970 Solar Terrestrial Physics 1, 99.

Colgate, S. A., Andouze, J. and Fowler, W. A. 1977 Ap. J. 213, 849.

Compton, A. H. and Getting, I. A. 1935 Phys. Rev. 47, 379.

Coppi, B. and Friedland, A. B. 1971 Ap. J. 169, 379.

Covington, A. E. 1948 Proc. I.R.E. 36, 454.

Cox, J. L. and Bennett, W. H. 1970 Phys.Fluids 13, 182.

Craig, I. J. D. and McClymont, A. N. 1976 Solar Phys. 50, 133.

Crannell, C. J., Frost, K. J., Matzler, C., Ohki, K. and Siba, J. L. 1977 "Impulsive Solar X-Ray Bursts", NASA GSFC Preprint X-684-202.

deFeiter, L. D. 1975 Solar Gamma-, X- and EUV Radiation, S. R. Kane, ed. (Dordrecht: Reidel), p. 185.

Deslandres, H. A. 1893 Compt. Rend. 117, 716.

Deslandres, H. A. 1905, Bull. Astron. 22, 332.

Donnelly, R. F. 1974 High Energy Phenomena on the Sun, R. Ramaty and R. G. Stone, eds. NASA SP-342, p. 242.

Dorman, I. 1974 Cosmic Rays: Variations and Space Exploration, (Cambridge: Cambridge University Press), p. 98.

Dum, C. T. and Dupree, T. D. 1970 Phys. Fluids 13, 2064.

Dungey, J. W. 1958 Cosmic Electrodynamics, (Cambridge: Cambridge University Press), p. 98.

Eddy, J. A. 1976 Science 193, 1189.

Eddy, J. A., Gilman, P. A. and Trotter, D. E. 1977 Science 198, 824.

Ekdahl, C., Greenspan, M., Kribel, R. E., Sethian, J. and Wharton, C. B. 1974 Phys. Rev. Lett. 33, 346.

Elcan, M. J. 1976 Bull. Am. Astr. Soc. 8, 556.

Evans, J. W. 1949 J. Opt. Soc. Am. 39, 229.

Friedman, M. and Hamberger, S. M. 1969 Solar Phys. 8, 104.

Frost, K. J. 1974 Coronal Disturbances, G. Newkirk, Jr., ed. (Dordrecht: Reidel), p. 421.

Frost, K. J. and Dennis, B. R. 1971 Ap. J. 165, 655.

Frost, K. J., Dennis, B. R. and Lencho, R. J. 1970 New Techniques in Space Astronomy, F. Labuhn and R. Lust, eds. (Dordrecht: Reidel), p. 185.

Giovanelli, R. G. 1946 Nature 158, 81.

Giovanelli, R. G. 1947 M.N.R.A.S. 107, 338.

Giovanelli, R. G. 1948 M.N.R.A.S. 108, 163.

Gold, T. and Hoyle, R. 1960 M.N.R.A.S. 120, 89.

Goldenbaum, G. C., Dove, W. F., Gerber, K. A. and Logan, B. G. 1974 Phys. Rev. Lett. 32, 830.

Graybill, S. E. and Nablo, S. V. 1966 Appl. Phys. Lett. 8, 18.

Hale, G. E. 1891 Sidereal Messenger 10, 257.

Hale, G. E. 1892 Astron. Astrophys. 11, 611.

Hale, G. E. 1906 Ap. J. 23, 92.

Hale, G. E. 1908 Ap. J. 28, 315.

Hale, G. E. 1929 Ap. J. 73, 379.

Hall, D. E. and Sturrock, P. A. 1967 Phys. Fluids 10, 2620.

Hall, L. A. 1971 Solar Phys. 21, 167.

Hammer, D. A. and Rostoker, N. 1970 Phys. Fluids 13, 1831.

Hey, J. S. 1946 Nature 157, 47.

Hodgson, R. 1859 M.N.R.A.S. 20, 15.

Holt, S. S. and Ramaty, R. 1969 Solar Phys. 8, 119.

Hoyle, F. 1948 Some Recent Researches in Solar Physics (Cambridge: Cambridge University Press), p. 93 ff.

Hoyng, P. 1977 Astron. Astrophys. 55, 23.

Hoyng, P., Brown, J. C. and Van Beek, H. F. 1976 Solar Phys. 48, 197.

Hoyng, P., Knight, J. W. and Spicer, D. S. 1978 "Diagnostics of Solar Flare Hard X-Ray Sources", Solar Phys. (in press).

Hudson, H. S. 1972 Solar Phys. 24, 414.

Hudson, H. S. 1974 High Energy Phenomena on the Sun, R. Ramaty and R. G. Stone, eds. NASA SP-342, p. 207.

Ionson, J. A. 1976 Phys. Letters 58A, 105.

Isaacson, E. and Keller, H. B. 1966 Analysis of Numerical Methods (New York: John Wiley).

Jansky, K. G. 1933 Proc. I.R.E. 21, 1387.

Janssen, P. J. 1869 Compt. Rend. 68, 713.

Kahler, S. 1975 Solar Gamma-, X-, and EUV Radiation, S. R. Kane, ed. (Dordrecht: Reidel), p. 211.

Kane, S. R. 1974 Coronal Disturbances, G. Newkirk, Jr., ed. (Dordrecht: Reidel), p. 105.

Kane, S. R. and Anderson, K. A. 1970 Ap. J. 162, 1003.

Kane, S. R. and Donnelly, R. F. 1971 Ap. J. 164, 151.

Kelly, P. T. and Rense, W. A. 1972 Solar Phys. 26, 431.

Kindel, J. M. and Kennel, C. F. 1971 J. Geophys. Res. 76, 3055.

Klok, O. D., Kumentsov, V. K., Strelkov, P. S. and Shkvarunets, A. G. 1974 Sov. Phys. JETP 40, 696.

Koch, H. W. and Motz, J. W. 1959 Rev. Mod. Phys. 31, 920.

Kostyuk, N. D. 1975 Sov. Astron. 19, 458.

Kostyuk, N. D. and Pikel'ner, S. V. 1975 Sov. Astron. 18, 590.

Kruger, A. 1972 Physics of Solar Continuum Radio Bursts (Berlin: Springer-Verlag).

Kundu, M. R. 1961 J. Geophys. Res. 66, 4308.

Langer, S. H. and Petrosian, V. 1977 Ap. J. 215, 666.

Lawson, J. D. 1957 J. Electron. Control 3, 587.

Lawson, J. D. 1958 J. Electron. Control 5, 146.

Lawson, J. D. 1959 J. Nucl. Energy Pt. C 1, 31.

Lee, R. and Sudan, R. N. 1971 Phys. Fluids 14, 1213.

Levine, L. S., Vitkovitsky, I. M., Hammer, D. A. and Andrews, M. L. 1971 J. Appl. Phys. 42, 1863.

Lin, R. P. 1974 Space Sci. Rev. 16, 189.

Lin, R. P. and Hudson, H. S. 1971 Solar Phys. 17, 412.

Lockyer, J. N. 1869 Phil. Trans. Roy. Soc. Lond. 159, 425.

Lovelace, R. V. and Sudan, R. N. 1971 Phys. Rev. Lett. 27, 1256.

Lyot, B. 1933 Compt. Rend. 197, 1593.

Meadows, A. J. 1970 Early Solar Physics (London: Pergamon).

Melrose, D. B. 1974 Solar Phys. 34, 421.

Miller, P. A. and Kuswa, G. W. 1973 Phys. Rev. Lett. 30, 958.

Millochau, G. and Stefanik, J. 1906, Ap. J. 24, 42.

Newman, C. E. 1973 J. Math. Phys. 14, 502.

Noyes, R. W. 1971 Ann. Rev. Astron. Astro. 9, 209.

Noyes, R. W. 1974 High Energy Phenomena on the Sun, R. Ramaty and R. G. Stone, eds. NASA SP-342, p. 231.

Moore, R. L. and Fung, P. C. W. 1972 Solar Phys. 23, 78.

Ohman, Y. 1938 Nature 141, 157.

Palmadesso, P. J., Coffey, T. P., Ossakow, S. L. and Papadopoulos, K. 1974 Geophys. Res. Lett. 1, 105.

Papadopoulos, K. 1977 Rev. Geophys. Space Phys. 15, 113.

Peterson, L. E., Datlowe, D. W. and McKenzie, D. L. 1974 High Energy Phenomena on the Sun, R. Ramaty and R. G. Stone, eds., NASA SP-342, p. 132.

Peterson, L. E. and Winckler, J. R. 1958 Phys. Rev. Lett. 1, 205.

Peterson, L. E. and Winckler, J. R. 1959 J. Geophys. Res. 64, 697.

Petrosian, V. 1973 Ap. J. 186, 291.

Priest, E. R. and Heyvaerts, J. 1974 Solar Phys. 36, 433.

Prono, D., Ecker, B., Bergstrm, N. and Benford, J. 1975 Phys. Rev. Lett. 35, 438.

Raymond, J. C. 1976 Private Communication.

Raymond, J. C., Cox, D. P. and Smith, B. W. 1976 Ap. J. 204, 290.

Reber, G. 1944 Ap. J. 100, 279.

Richtmyer, R. D. and Morton, K. W. 1967 Difference Methods for Initial Value Problems (New York: John Wiley).

Roberts, T. G. and Bennett, W. H. 1968 Plasma Phys. 10, 381.

Rust, D. M. 1977 "Solar Flares" in Solar System Plasma Physics, C. F. Kennel, L. J. Lanzerotti and E. N. Parker, eds. (in press).

Scheiner, C. 1630 Rosa Ursina sive Sol ex Admirando Facularum et Macularum. . . (Bracciano).

Schwabe, H. 1844 Astron. Nach. 21, 233.

Secchi, A. 1877 Le Soleil, (Paris: Gauthier-Villars).

Shapiro, P. R. and Knight, J. W. 1978 "The Rapid Heating of Coronal Plasma During Solar Flares: Nonequilibrium Ionization Diagnostics and Reverse Currents", Ap. J. (in press).

Smith, D. F. 1974 Coronal Disturbances, G. Newkirk, Jr., ed. (Dordrecht: Reidel), p. 253.

Smith, D. F. 1975 Ap. J. 201, 521.

Smith, D. F. 1977a Ap. J. 212, 891.

Smith, D. F. 1977b Ap. J. 217, 644.

Smith, D. F. and Lilliequist, C. G. 1978 "Thermal versus Nonthermal Interpretations of Solar Hard X-ray Emission" Ap. J. (submitted).

Smith, E. V. P. and Gottlieb, D. A. 1974 Space Sci. Rev. 16, 771.

Smith, H. J. and Smith, E. V. P. 1963 Solar Flares (New York: Macmillan).

Southworth, G. C. 1945 J. Franklin Inst. 239, 285.

Spicer, D. S. 1977 Solar Phys. 53, 305.

Spitzer, L. 1962 Physics of Fully Ionized Gases (New York: John Wiley).

Sturrock, P. A. 1966 Phys. Rev. 141, 186.

Sturrock, P. A. 1968 Structure and Development of Solar Active Regions, K. O. Kuipenheuer, ed. (Dordrecht: Reidel), p. 471.

Sturrock, P. A. 1974 Coronal Disturbances, G. Newkirk, Jr., ed. (Dordrecht: Reidel), p. 437.

Sturrock, P. A. and Coppi, B. 1966 Ap. J. 143, 3.

Svestka, Z. 1975 Solar Flares, (Dordrecht: Reidel).

Sweet, P. A. 1958 Electromagnetic Phenomena in Cosmical Physics, B. Lehnert, ed. (Cambridge: Cambridge University Press), p. 123.

Sweet, P. A. 1969 Ann. Rev. Astron. Astro. 6, 149.

Syrovatsky, S. I. 1969 Solar Flares and Space Research, C. deJager and Z. Svestka, eds. (Dordrecht: Reidel), p. 396.

Takakura, T. 1972 Solar Phys. 26, 151.

Takakura, T. and Kai, K. 1966 Pub. Astron. Soc. Japan 18, 57.

Tanaka, H. and Nagakawa, Y. 1973 Solar Phys. 33, 187.

Wolf, R. 1852 Compt. Rend. 35, 704.

Yonas, G. and Spence, P. 1969 Proc. 10th Symp. on Electron, Ion and Laser Beam Tech., L. Marton, ed. (San Francisco: San Francisco Press), p. 143.

Young, C. A. 1871 Am. J. Sci. 102, 468.

Young, C. A. 1902a The Sun (New York: Appleton), p. 1.

Young, C. A. 1902b The Sun (New York: Appleton), pp. 166-167.

Young, C. A. 1902c The Sun (New York: Appleton), p. 225.