UNIVERSITY OF ILLINOIS
URBANA

AERONOMY REPORT NO. 113

ROCKET OBSERVATIONS OF THE IONOSPHERE DURING THE ECLIPSE OF 26 FEBRUARY 1979

by
M. K. McInerney
L. G. Smith

June 1, 1984

Library of Congress ISSN 0568-0581

Aeronomy Laboratory
Department of Electrical and Computer Engineering
University of Illinois
Urbana, Illinois

Supported by
National Aeronautics and Space Administration

UILU-ENG-84-2501

A E R O N O M Y R E P O R T

N O. 113

ROCKET OBSERVATIONS OF THE IONOSPHERE DURING THE ECLIPSE OF 26 FEBRUARY 1979

by

M. K. McInerney
L. G. Smith

June 1, 1984

Supported by
National Aeronautics
Space Administration
Grant NGR 14-005-181

Aeronomy Laboratory
Department of Electrical and
Computer Engineering
University of Illinois
Urbana, Illinois

ABSTRACT

Electron density profiles and energetic particle fluxes have been determined from two rockets launched, respectively, at the beginning and end of totality during the solar eclipse of 26 February 1979. These, and one other rocket at the same time of day on 24 February 1979, were launched from a temporary site near Red Lake, Ontario. The electron density profile from 24 February 1979 shows the electron density to be normal (at 1×10^5 cm^{-3}) above 110 km, to rocket apogee (130.5 km). Below 110 km the electron density is enhanced, by an order of magnitude in the D-region, compared with data from Wallops Island at the same solar zenith angle (63°). The enhancement is qualitatively explained by the large flux of field-aligned energetic particles (mainly electrons) observed on the same rocket. During totality, on 26 February 1979, the electron density above 110 km to rocket apogee (132.6 to 132.3 km) is reduced by a factor of about three, as seen in other eclipses. Below 110 km, however, the electron density is much greater than observed during previous eclipses. Again this is attributed to the additional ionization due to energetic particles. The particle flux measured on the 26 February was an order of magnitude less than that on the 24 February but showed greater variability, particularly at the higher energies (100 keV). A feature of the particle flux is that, for the two rockets that were separated horizontally by 38 km while above the absorbing region, the variations are uncorrelated.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

1. INTRODUCTION

A solar eclipse is of great interest to scientists. In particular it gives aeronomers the opportunity to study the interaction of solar radiation with the atmosphere. Relative to the day-night transition the eclipse gives a rapid change; the total duration is about 2 hours. This allows the short-term response of the ionosphere to be studied.

The occurrence of the total solar eclipse of 26 February 1979 across North America (mainly in Canada) provided a convenient opportunity for a campaign of sounding rocket launches supported by ground experiments. Table 1.1 lists the sounding rocket measurements of greatest interest.

The Aeronomy Laboratory of the University of Illinois collaborated with the University of Bern, Switzerland, under NASA sponsorship, in three rocket launches. Three experiments (in each payload) of the University of Illinois are the subject of this report. They are identified as probe, propagation and particle experiments, respectively.

The particle experiment was included to protect against the remote possibility of confusion within the eclipse effects caused by a particle precipitation event. As it happens there was such an event during the eclipse (and during the pre-eclipse launch of 24 February). The particle data have proved to be valuable, however, providing a unique data set on a daytime aurora.

This introductory chapter details the eclipse circumstances and outlines this report.

1.1 Eclipse Circumstances

The three University of Illinois rockets launched for the study of the total solar eclipse of 1979 were Nike Tomahawks and were similarly instrumented. The only difference between the three payloads was the type

Table 1.1 Sounding rocket measurements of particular interest.

Launch Date	Vehicle Number	Launch Time (UT)	Approx. Apogee (km)	Approx. Flt. Time (sec)	Measured Parameters
2/24/79	18.1020UE	1652	137	870	Positive Ion Composition; Electron Density; Particle Flux
2/26/79	A-1	1628	133	354	Electron Density
2/26/79	B-1	1628:30	150	374	Neutral Density; Electron Density; Particle Flux
2/26/79	AMF-VA-51	1650:45	135	700	Electron Density; Positive Ion Density
2/26/79	23.009UE	1650:50	82	6300	Positive and Negative Conductivity
2/26/79	18.1021UE	1652	137	870	Positive Ion Composition; Electron Density; Particle Flux
2/26/79	C-1	1652:30	120	700	Positive and Negative Ion Composition; Ion Density
2/26/79	33.004UE	1653:30	194	700	Electron Density
2/26/79	18.1022UE	1654:10	137	870	Negative Ion Composition; Electron Density; Particle Flux
2/26/79	C-2	1741	120	700	Positive and Negative Ion Composition; Ion Density
2/27/79	23.010UE	1200	82	6300	Positive and Negative Conductivity

of mass spectrometer operated by the University of Bern, Switzerland.

Two of the rockets were launched during the total solar eclipse and one two days prior (to serve as a reference). Nike Tomahawk 18.1020 was launched at 1652:00 UT on 24 February 1979 and carried a positive ion mass spectrometer. Nike Tomahawks 18.1021 and 18.1022 were launched during the eclipse on 26 February 1979: 18.1021 at 1652:00 UT and 18.1022 at 1654:10 UT. 18.1021 carried a positive ion mass spectrometer while 18.1022 carried a negative ion mass spectrometer.

The launch of 18.1021 occurred just after second contact. The launch of 18.1022 occurred just before third contact.

The launch site was located near the town of Red Lake in Northwest Ontario. The geographic coordinates of the site are 50.9007°N and 93.4538°W. The instrumentation site, at which were located the NASA telemetry systems and the U of I equipment van and transmitting antennas. was situated about five miles from the launch site. Figure 1.1 shows the position of the launch site in relation to the main highways in the area.

The rockets' trajectories were close to the predicted trajectories with apogees of 132.6 km and 132.3 km for Nike Tomahawks 18.1021 and 18.1022, respectively. Apogee of the pre-eclipse launch (Nike Tomahawk 18.1020) was 130.5 km.

The trajectories of 18.1021 and 18.1022 are shown in Figure 1.2. The eclipse circumstances at the position of each rocket are indicated along its trajectory. For times when a rocket is outside totality the number given is the percentage of the disc that is visible. Where a rocket is in totality the time since second contact is given.

The first rocket enters totality at T+60 sec, at an altitude of 61 km on ascent and exits at T+300 sec, at an altitude of 68.5 km on descent.

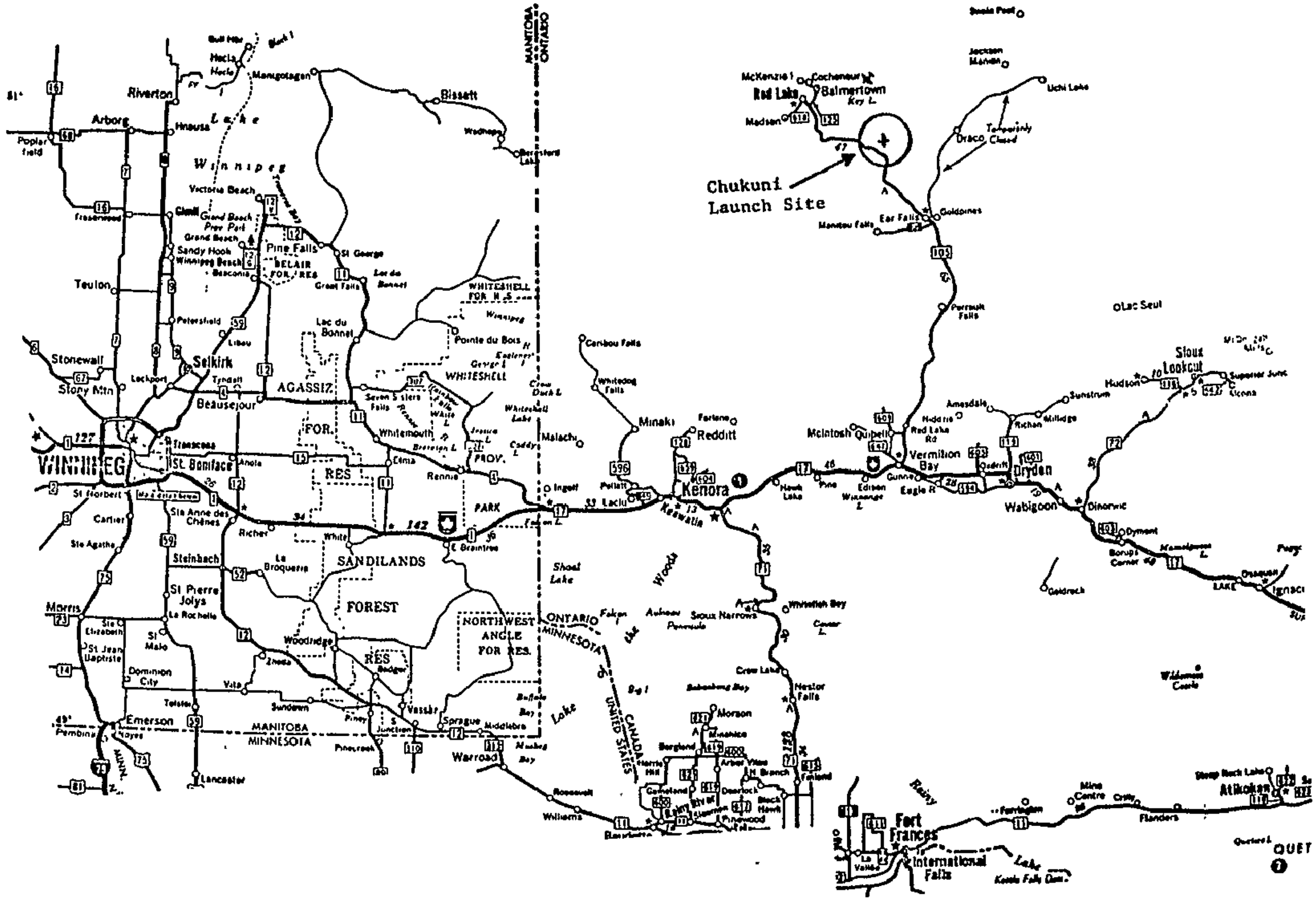


Figure 1.1 Red Lake area map.

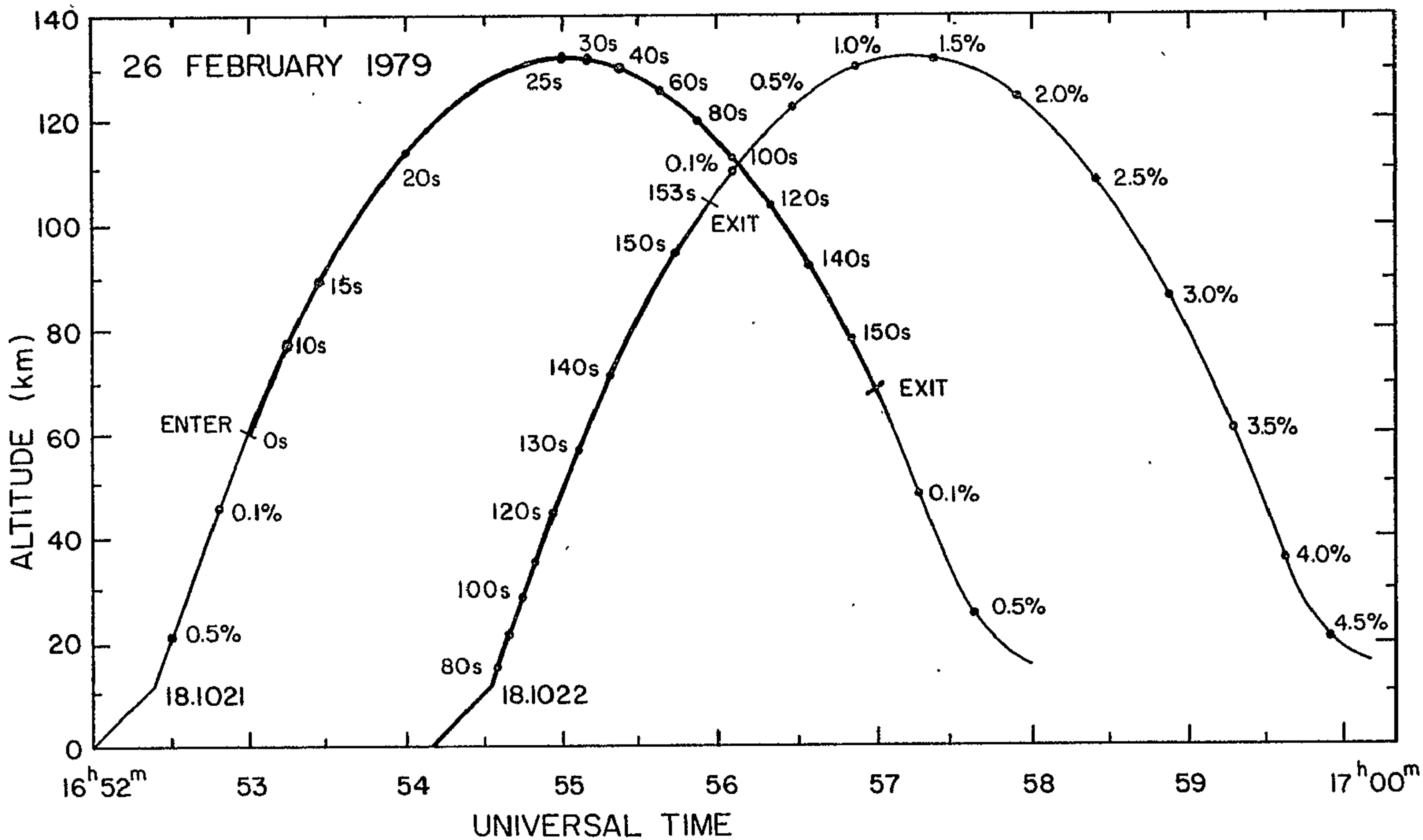


Figure 1.2 Eclipse circumstances at the position of the rocket. Marked along each rocket trajectory is the percentage of the solar disc that is visible and, inside totality, the time (seconds) since second contact.

The second rocket remains in totality from launch until T+106 sec, at an altitude of 105 km on ascent.

Another perspective of the eclipse circumstances is shown in Figure 1.3. The sun-moon distance, in solar radii, is plotted against time. The magnitude of the eclipse was 1.0403 (i.e., the ratio of the apparent diameter of the moon and the sun) so that a sun-moon distance of 0.0403 solar radii represents the edge of totality and zero the center of totality. This figure is a picture of the rocket position in terms of radial distance from the center of totality.

A complementary view of the trajectories is shown in Figure 1.4. This is a polar plot of the sun-moon distance versus angular position relative to the sun's North point, and gives a picture of the rocket position viewed along the axis of the shadow.

1.2 Outline

The major concern of this report is the processing of data from three rocket flights for the solar eclipse of 1979. A brief description of the effects that a solar eclipse and energetic particles have on the lower ionosphere is given to provide background for the experiments.

The first part of this introductory chapter described the details of the rocket flights (e.g., launch location, launch time, apogee). For the two rockets launched in totality, various trajectory plots are presented. Two of the plots show the rockets' locations relative to the eclipse shadow.

Chapter 2 briefly describes the major effects which a solar eclipse has on the lower ionosphere. The D- and E-region effects are described separately because of their differing chemistries. (Negative ions can be neglected in the E-region.) Two important ionospheric parameters, the effective recombination coefficient and the $[NO^+]/[O_2^+]$ ratio, are

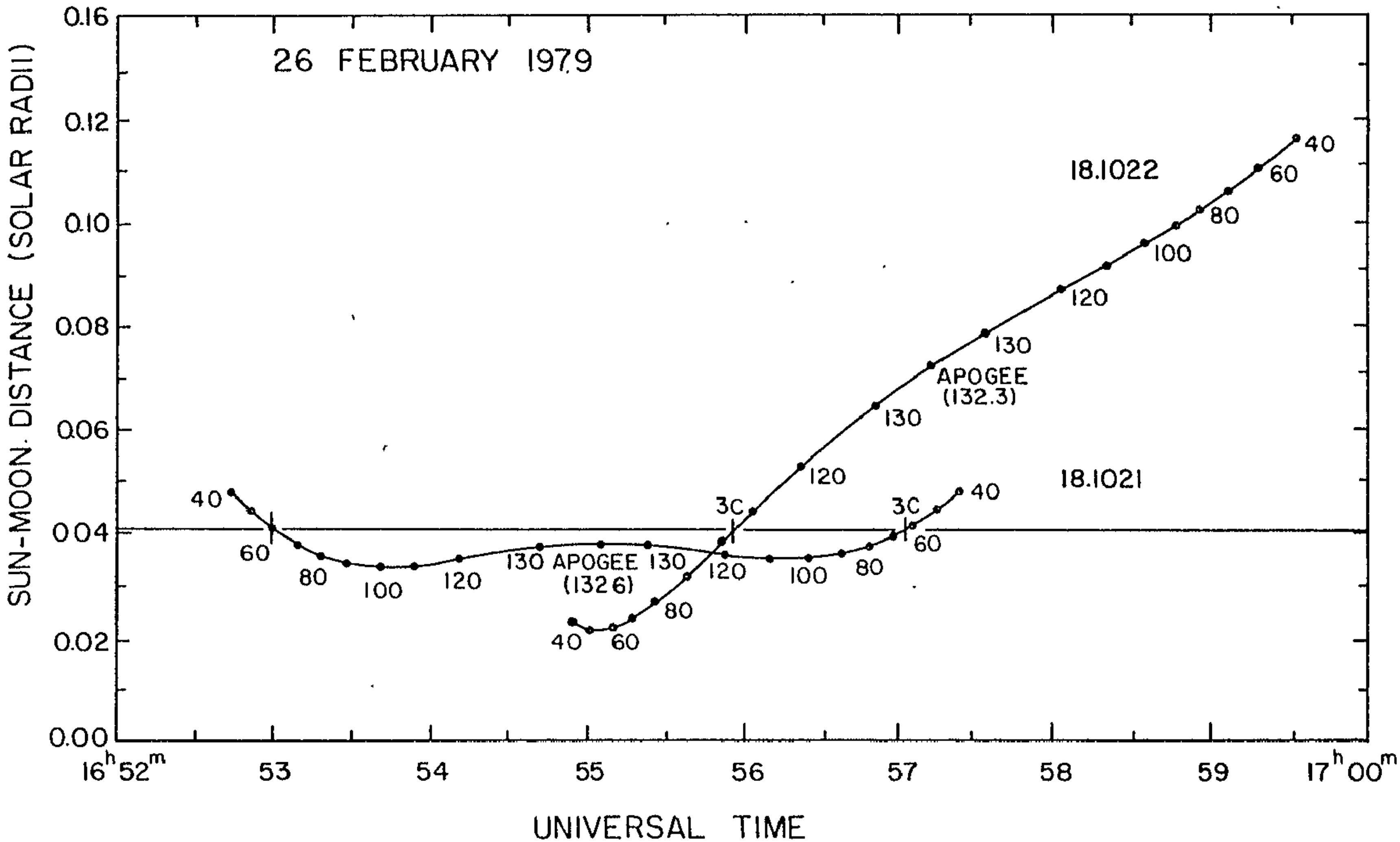


Figure 1.3 The eclipse circumstances at the position of the rocket represented by the radial distance from the shadow axis. The rocket altitude (km) is marked along each curve.

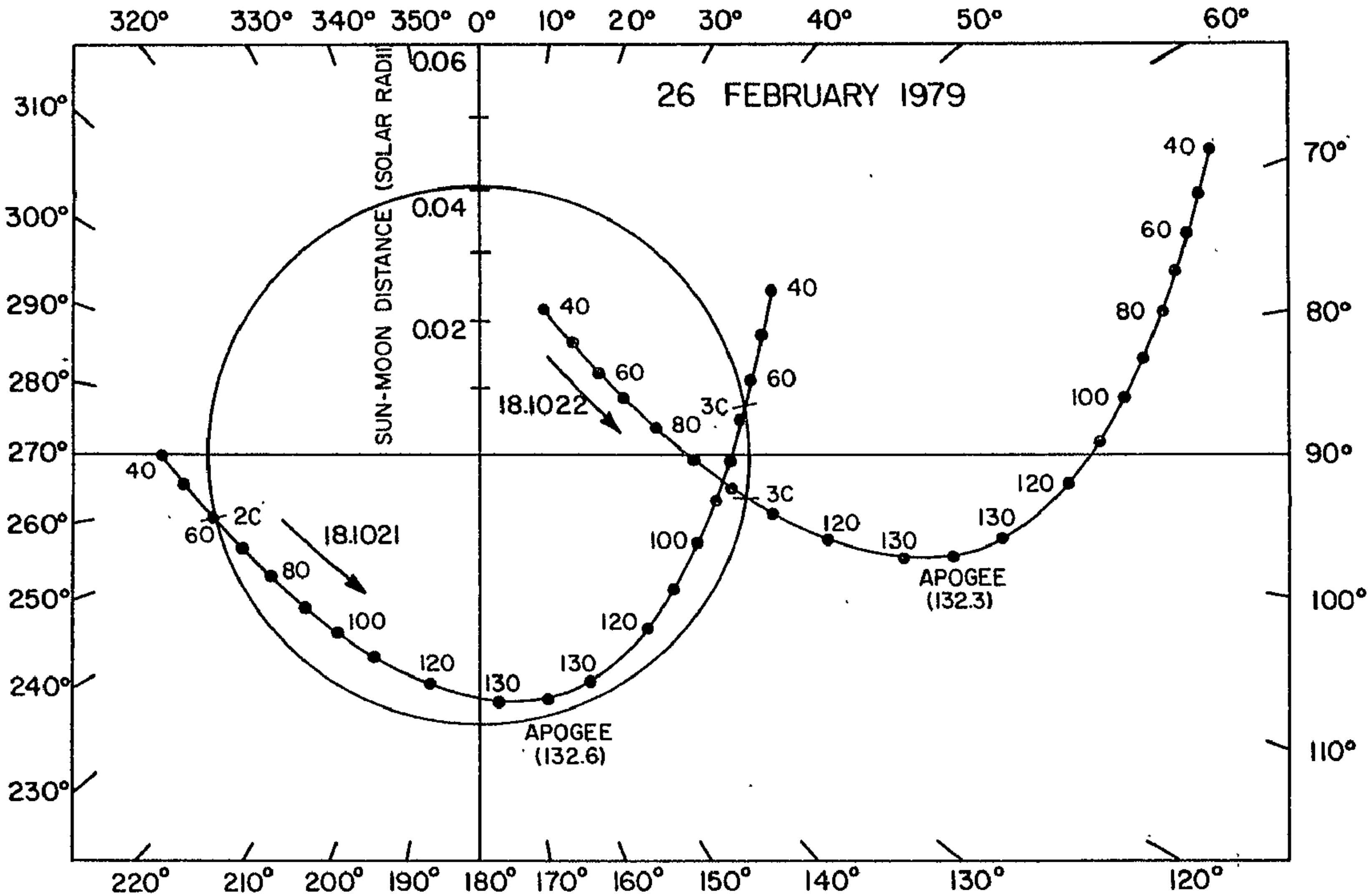


Figure 1.4 The eclipse circumstances here show the rocket position viewed along the axis of the shadow. The rocket altitude (km) is marked along each curve.

discussed.

The effects of precipitating electrons on the lower ionosphere are summarized in Chapter 3. The major effect, atomic emissions, is discussed. The effect on electron concentration and the $[NO^+]/[O_2^+]$ ratio are also discussed.

Chapter 4 presents detailed descriptions of the probe, propagation and energetic particle experiments. First the theory upon which each experiment is based is summarized, then the experiment is described in detail. Numerous references are given to aid in understanding the theory and to document the experimental method.

The data processing techniques are detailed in Chapter 5. Each computer program used in the analysis is described. For each experiment the computer programs are presented in the order that they are executed. Data from flight 18.1020 are used as an example. In most cases the data requires manual processing between programs. Manual processing steps are described with the computer program for which they produce input. Some of the experiments are continually undergoing changes, making a detailed program description unnecessary. In these cases the programs are merely outlined.

Chapter 6 presents the final results of each of the experiments for all three rocket flights. The determination of electron density using the propagation experiment is addressed briefly. A short discussion of the electron concentration and energetic particle profiles is included. Comparisons to previous rocket flights are made. Finally, a few topics are suggested for further study.

The bulk of this thesis is the appendices. Each computer program is listed along with a sample of the input parameters necessary for execution. For programs which run on the IBM, the JCL required for compilation and execution is given. For programs which run on the Cyber, the Cyber Operating System commands required for compilation and execution as a batch job are given.

Two programs, SWEEP and EDPLOT, are designed to run interactively on the Cyber. In these cases the command file given must be entered while in the time sharing mode.

2. SOLAR ECLIPSE EFFECTS IN THE LOWER IONOSPHERE

2.1 Introduction

The word eclipse comes from the Greek ekleipsis "abandonment". The sun is eclipsed when the moon comes between it and the earth so that the moon's shadow sweeps over the earth. This shadow consists of two parts: the umbra, or total shadow, a cone in which there is no direct sunlight; and the penumbra, or partial shadow, which is reached by light from only a part of the sun's disk.

The apparent diameters of the sun and moon as seen from the earth's surface are nearly equal (about 0.5°). Since the earth is in an elliptical orbit about the sun, the apparent size of the sun changes slightly during a year. Similarly, the moon is in an elliptical orbit about the earth and therefore the moon's apparent size changes slightly during a month. When the sun is nearest to the earth and the moon is at its greatest distance, the apparent disk of the moon is smaller than that of the sun. If an eclipse were to occur at this time, the moon's disk passing over the sun's disk would not cover it completely but would leave the rim of the sun visible all around it. This is an "annular" solar eclipse. A "total" solar eclipse occurs when the moon's disk completely covers the sun's disk. Totality during any particular solar eclipse can only be seen from a narrow belt on the earth.

There are various phases of a total solar eclipse, separated by "contact" times. "First contact" designates the moment when the disk of the moon just touches the disk of the sun: the partial phase of the eclipse begins. The disk of the moon now moves slowly across the sun's disk, and the bright area of the sun is reduced to a crescent. At the moment of second contact the sun vanishes and totality begins. The maximum duration of totalilty is 7 min 40 sec and occurs at the equator. "Third contact" marks the beginning of the second partial phase. At third contact the sun reappears, as the moon's disk now gradually uncovers the sun's disk. "Fourth contact" represents the end of the total solar eclipse, the moon's disk now no longer blocking the sun's disk. Since the path of a total solar eclipse is on the average only 100 km wide, a given location on earth will, on average, experience totality only once in 360 years. (BALDWIN (1965) covers the geometry of eclipse situations.)

A solar eclipse reduces the amount of solar radiation reaching the earth's atmosphere. The rest of this chapter will discuss eclipse effects, with emphasis on the ionospheric D and E regions.

2.2 Solar Radiation

The sun emits energy at a rate of 3.86×10^{33} erg/sec of which only one part in 2.2×10^{9} intercepts the earth. The visible surface of the sun is the photosphere. Above it a layer 5,000 km thick constitutes the inner atmosphere, or chromosphere, while above this lies the very tenuous high-temperature corona. Each region of the sun differs in chemical composition and temperature distribution and hence in the spectra of emitted radiation.

As the sun's disk is covered by the moon's disk during a solar eclipse, the amount of solar radiation reaching the earth is progressively reduced. The ratio of the instantaneous solar flux at a given wavelength to its uneclipsed value is called the "eclipse function". For visible radiation during a total eclipse, it is, by definition, unity at first and fourth contact, and zero at second and third contact. The eclipse function for visible radiation is almost equal to the fraction of the solar disc that is not obscured.

Eclipse functions for wavelengths in the ultraviolet or X-ray region are not so well known. Regions of increased brightness (of UV and X-radiation) on the solar disk may cause the eclipse function to vary irregularly with solar obscuration; and radiation from outside the visible disk may give an eclipse function which does not become zero during totality.

As an example of non-uniform solar radiation, Figure 2.1 (SEARS, 1972) gives the obscuration functions calculated for visible light and ultraviolet (UV) radiation measured for Lyman-α (L_α) and X-radiation for the 1966 solar eclipse. For ultraviolet, Lyman-α and visible radiation the eclipse function decreased uniformly with time, virtually reaching zero at totality. However, X-radiation (which produces ionization in the D- and lower E-regions) only reached a minimum of about 20%, demonstrating that the solar disk is non-uniform in brightness.

SMITH (1972) measured Lyman-α (121.6 nm), an important ionization source of the D-region, during the eclipse of 7 March 1970. He found that at the center of totality the flux of Lyman-α from the solar corona is 0.15% of the flux from the unobscured sun. The flux at second contact is 0.64%; at third contact two observations give 0.52% and 0.59%, respectively.

During the same eclipse ACCARDO ET AL. (1972) made measurements of solar X-rays in the bands 0.2-0.8 nm, 0.8-2 nm and 4.4-6 nm (an important source of E-region ionization). The residual flux at totality was found to be 5, 7 and 16%, respectively, of the flux from the uneclipsed sun.

2.3 E Region

2.3.1 Effective recombination coefficient. The E-region is ionized by radiation in the spectral region greater than 1 nm. The ionization threshold for nitrogen and oxygen in atomic and molecular form, falls in the range 80-102 nm. The major loss process for E-region ionization is

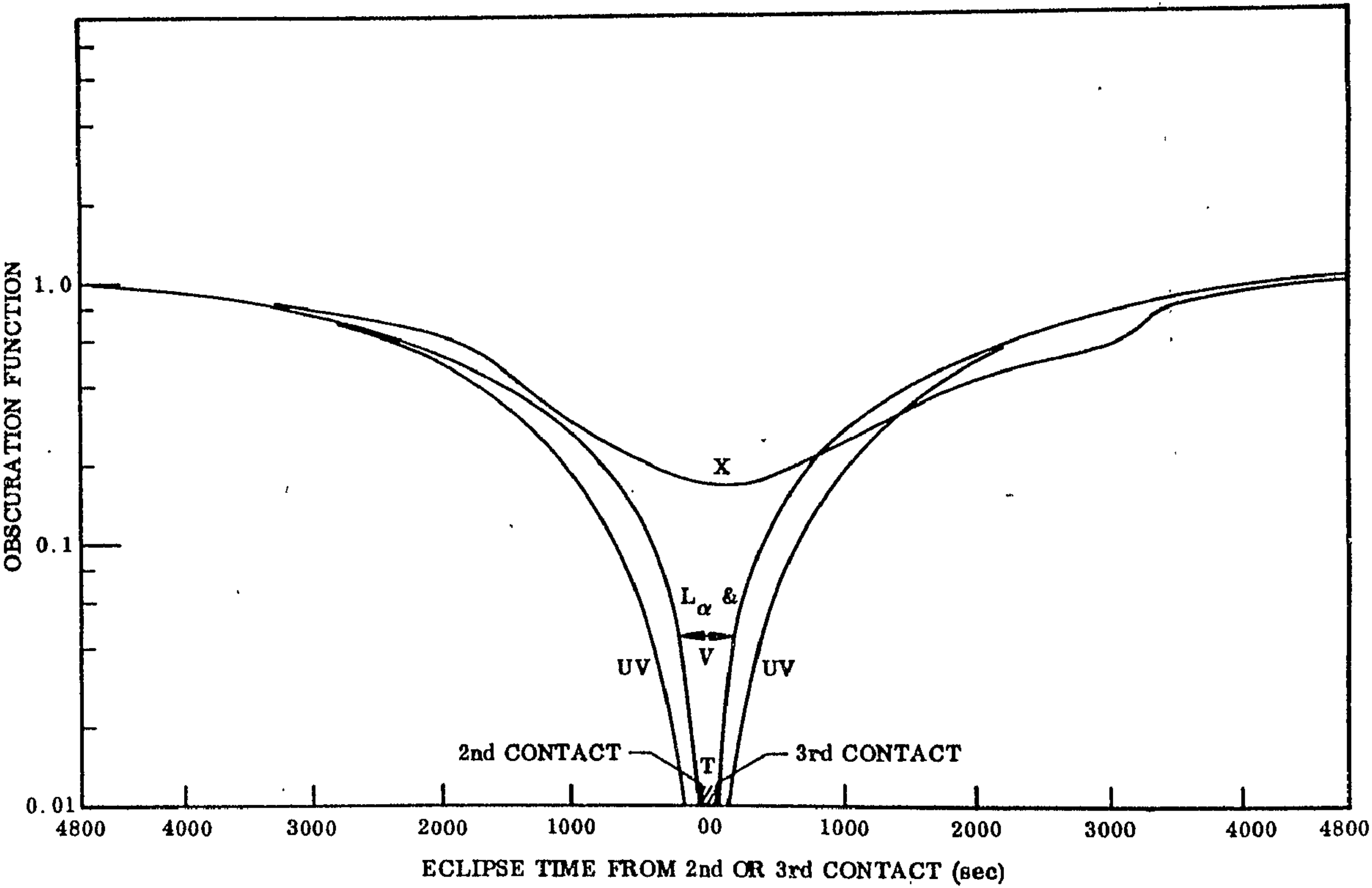


Figure 2.1 Obscuration functions for visible light (V), Lyman alpha (Lα), ultraviolet (UV), and X-ray (X) ionizing fluxes for the 1966 solar eclipse (SEARS, 1972). The obscuration functions for Lα and X were measured while the obscuration functions for V and UV were calculated.

dissociative recombination of molecular ions (O_2^+, NO^+, N_2^+) with electrons. If the E-region is modeled using a single species then, under steady-state conditions production and loss rates are equal, giving

$$q = \alpha_e N_e^2 \tag{2.1}$$

where q is the production function, α_e is the effective recombination coefficient and N_e is the electron density (RISHBETH ET AL., 1969). For this equation to hold the concentration of molecular ions must be much greater than the concentration of atomic ions; a situation that exists in the daytime E-region.

Early attempts in E-region eclipse analyses assumed that E-region electron production, q, was proportional to the uneclipsed area of the visible disk, and by comparing the assumed variation in the production function with the observed variation in electron density throughout an eclipse, values of α_e could be determined. Figure 2.2 (after SZENDREI AND MCELHINNY, 1956) illustrates typical E-region electron density variations during an eclipse. The minimum of electron density occurs close to the time of second and third contact, and is about half that on a normal day. Also shown are solutions to the continuity equation (equation 2.1) for different values of α_e and assuming q proportional to the eclipse function. The problem with these results was that: (1) if α_e were chosen to be about $10^{-8} cm^3 sec^{-1}$ such that the model electron density matched the observed density at totality, then the model electron density curve would have about a 17 min lag with respect to the observed electron density curve, and (2) if α_e were taken to be about $10^{-7} cm^3 sec^{-1}$ such that the model density curve had no delay relative to the observed curve, then the model density at totality was much lower than the observed value.

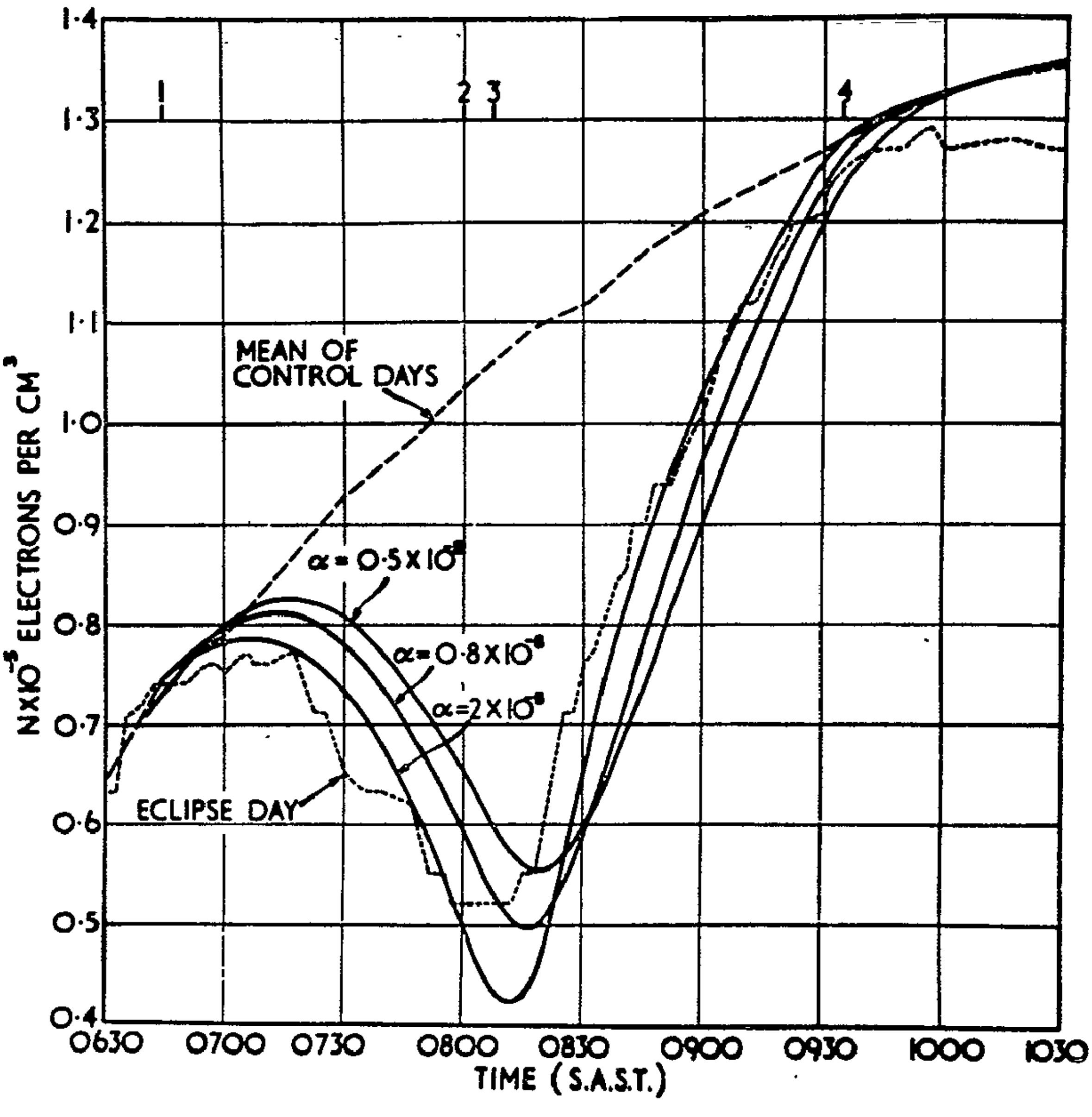


Figure 2.2 Theoretical and experimental peak electron densities for the E region during an eclipse (after SZENDREI AND MCELHINNY, 1956).

This problem was not resolved until laboratory measurements of α_e were made and solar radiation spectra during an eclipse were observed. It is now known that the recombination coefficients for NO and O_2 are 5×10^{-7} and $2 \times 10^{-7} cm^3 sec^{-1}$, respectively. So if the radiation ionizing the E-region is not fully eclipsed at totality, then a high recombination coefficient on the order of $10^{-7} cm^3 sec^{-1}$ will prevent a large time lag, and the residual radiation will account for the high residual electron density at totality (BOWHILL, 1969). This has been verified by rocket results during eclipses indicating that 10% to 20% of the X-radiation from the sun remains at totality (ACCARDO ET AL., 1972).

2.3.2 <u>$[NO^+]/[O_2^+]$ ratio.</u> Figures 2.3, 2.4 and 2.5 (from NARCISI ET AL., 1972) show the major positive ions in the lower E-region. The ions with masses 30 and 32 amu are NO^+ and O_2^+ respectively. Above 90 km, although the total ion concentrations are believed to be quite accurate, an error of ±50% is assigned to the absolute concentration of the individual ions.

In order to understand how the $[NO^+]/[O_2^+]$ ratio is expected to change during an eclipse, it is necessary to examine the NO^+ and O_2^+ production and loss processes. The most important chemical reactions involving these ions are:

$$O^+ + O_2 \rightarrow O_2^+ + O \qquad k_1 \tag{2.2}$$

$$O^+ + N_2 \rightarrow NO^+ + N \qquad k_2 \tag{2.3}$$

$$N_2^+ + O \rightarrow NO^+ + N \qquad k_3 \tag{2.4}$$

$$N_2^+ + O_2 \rightarrow O_2^+ + N_2 \qquad k_4 \tag{2.5}$$

$$O_2^+ + NO \rightarrow NO^+ + O_2 \qquad k_5 \tag{2.6}$$

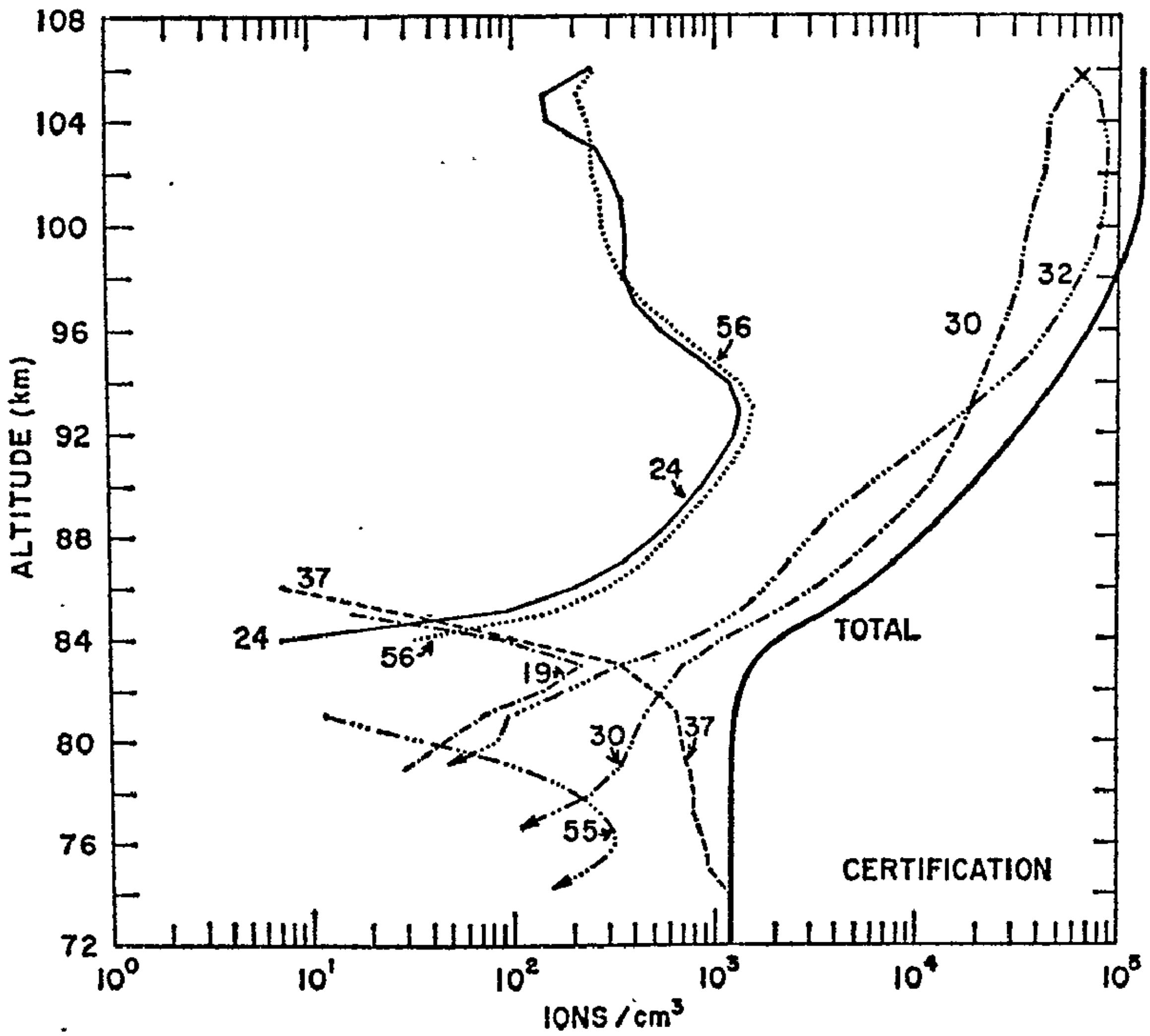


Figure 2.3 Major positive ions in the D and E regions for a full sun at about 20° solar zenith angle (NARCISI ET AL., 1972).

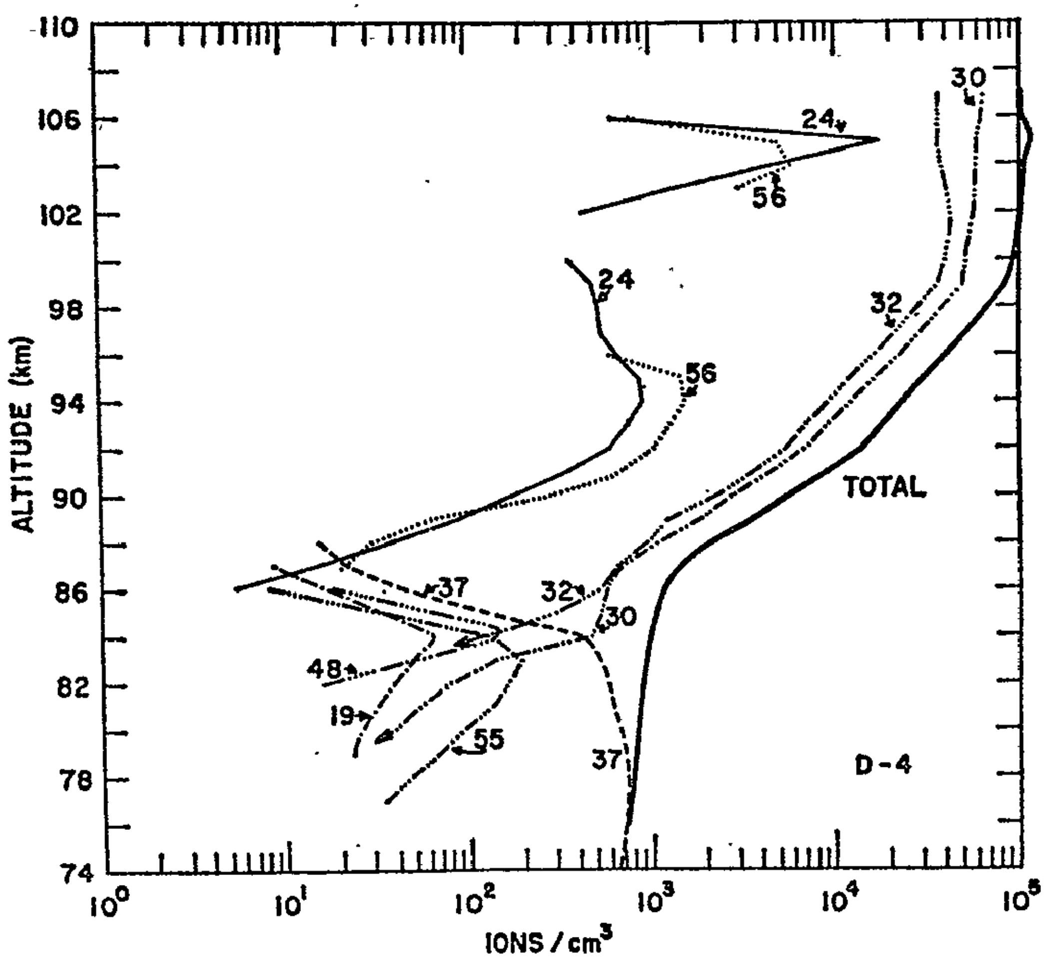


Figure 2.4 Major positive ions in the D and E regions. The rocket was rising at 90 km when the sun was 80 per cent obscured (NARCISI ET AL., 1972).

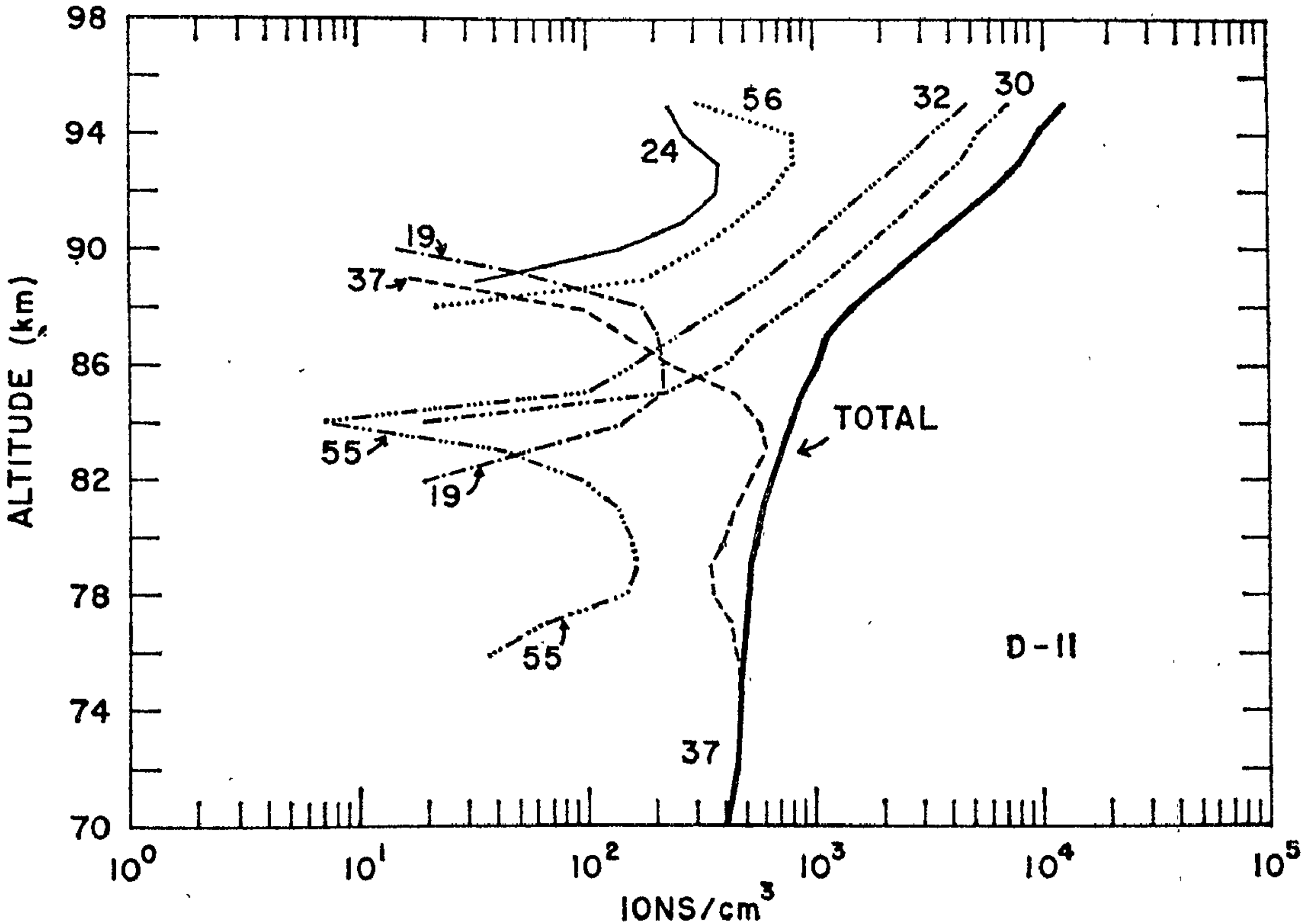


Figure 2.5 Major positive ions in the D and E regions at totality. At the trajectory altitudes of 70 km (ascent), 95 km (apogee) and 70 km (descent), the elapsed times after second contact were about 40 sec, 80 sec and 115 sec, respectively (NARCISI ET AL., 1972).

where k_n is the rate coefficient of the reaction. To derive an expression for the ratio $[NO^+]/[O_2^+]$ let the photoionization production rate for O_2^+ (i.e., $O_2 + h\nu \rightarrow O_2^+ + e$) be q_{O_2} and the recombination coefficient for NO^+ be α_1. Further, assume that recombination of O_2^+ with electrons is negligible. (For a discussion of E-region processes, see SWIDER (1972)). Equating production and loss in the steady-state yields

$$\alpha_1[e][NO^+] = k_2[O^+][N_2] + k_3[N_2^+][O] + k_5[O_2^+][NO] \qquad (2.7)$$

and

$$k_5[NO][O_2^+] = q_{O_2^+} + k_1[O^+][O_2] + k_4[N_2^+][O_2] = Q_{O_2^+} \qquad (2.8)$$

whence

$$\frac{[NO^+]}{[O_2^+]} = \frac{k_5[NO]}{\alpha_1[e]}\left(\frac{k_2[O^+][N_2] + k_3[N_2^+][O] + k_5[O_2^+][NO]}{q_{O_2^+} + k_1[O^+][O_2] + k_4[N_2^+][O_2]}\right) \qquad (2.9)$$

or

$$\frac{[NO^+]}{[O_2^+]} = \frac{k_5[NO]}{\alpha_1[e]}\left(\frac{k_2[O^+][N_2] + k_3[N_2^+][O] + Q_{O_2^+}}{Q_{O_2^+}}\right) \qquad (2.10)$$

where $Q_{O_2^+}$ is given by 2.8.

To see how the ratio $[NO^+]/[O_2^+]$ is expected to vary during the eclipse, notice that the various production term variations should cancel and the total variation should be controlled by the $1/[e]$ factor. Thus if all O_2^+ is lost to NO^+, the ratio $[NO^+]/[O_2^+]$ should increase as $1/[e]$ since the NO^+ loss rate varies as $[e]$ and the O_2^+ loss rate is independent of $[e]$ (OLIVER, 1973).

The concentrations shown in Figures 2.3, 2.4 and 2.5 show that the $[NO^+]/[O_2^+]$ ratio increases with progression into totality, although this can be questioned, considering the error placed on the results. However, E-region positive ion measurements reported by BRACE ET AL. (1972) show a definite increase in the $[NO^+]/[O_2^+]$ ratio in totality.

Also, some increase in the $[NO^+]/[O_2^+]$ ratio should be expected due to selective decreases in solar radiation. The chromospheric 91.1 to 102.7 nm band, which ionizes only O_2, is more effectively eclipsed than the coronal X-rays, which ionize all constituents.

2.4 D Region

2.4.1 Ionization sources. The photoionization of NO by solar Lyman-α radiation (121.2 nm) is a major source of NO^+ and free electrons in the daytime D region. Figure 2.6 (THOMAS, 1974) illustrates the relative roles of solar Lyman-α, solar UV, hard X-rays (0.2-0.8 nm), and galactic cosmic rays (GCR) as sources of ionization in the daytime D-region below 90 km. Above 90 km solar Lyman-β, solar EUV (extreme ultraviolet) and soft X-rays (3.1-10 nm) become major sources of positive ions such as N_2^+ and O_2^+. Generally, the X-rays and GCR produce N_2^+ and O_2^+ ions. The N_2^+ ions are quickly converted via charge exchange to O_2^+ ions, which in turn may react with NO, electrons, or O_2 depending on relative concentrations and reaction rate constants, which vary with altitude.

$O_2(^1\Delta_g)$ is a major source of O_2^+ in the daytime D region below 90 km. It is ionized by UV radiation in the 102.7-111.8 nm band. Figure 2.7 gives O_2^+ production rates for quiet-sun conditions. Illustrated are the relative importance of Lyman-β, hard and soft X-rays, and 102.7-111.8 nm solar radiations in the photoionization of O_2 and $O_2\ (^1\Delta_g)$.

Precipitating electrons are another possible source of NO^+ and free electrons in the D region. However, their role will be discussed in the next chapter.

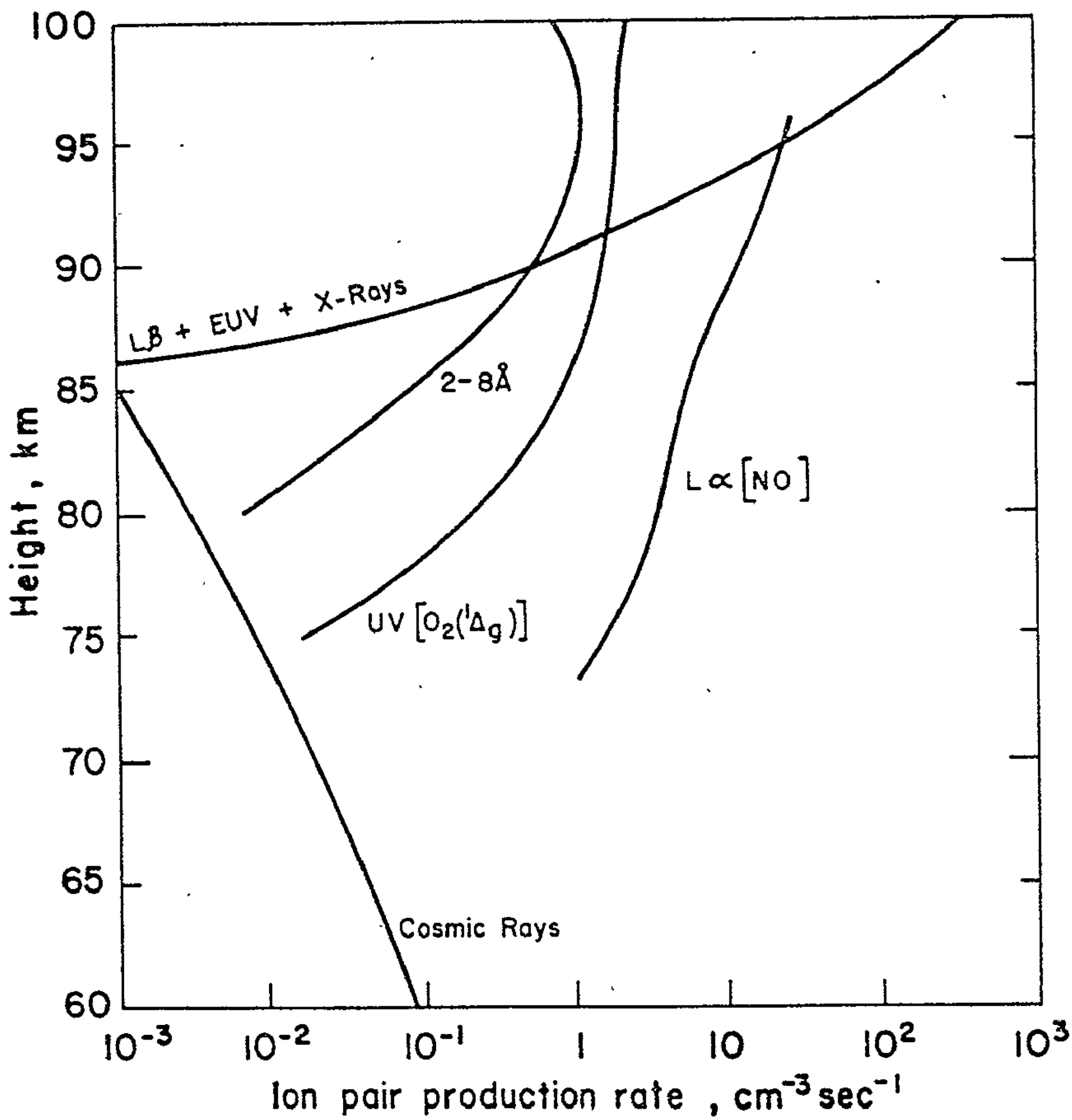


Figure 2.6 Ionization rates in the quiet daytime D region for solar minimum conditions and solar zenith angles near 60°. [NO] represents the effect of Lyman-α radiation on nitric oxide, $UV[O_2(^1\Delta_g)]$ the effect of 102.7-111.8 nm radiation on metastable oxygen molecules, Lβ + EUV + X-rays the combined effect of Lyman-β, EUV, and soft X-ray radiation, and 2-8 Å (0.2-0.8 nm) the effect of hard X-rays (THOMAS, 1974).

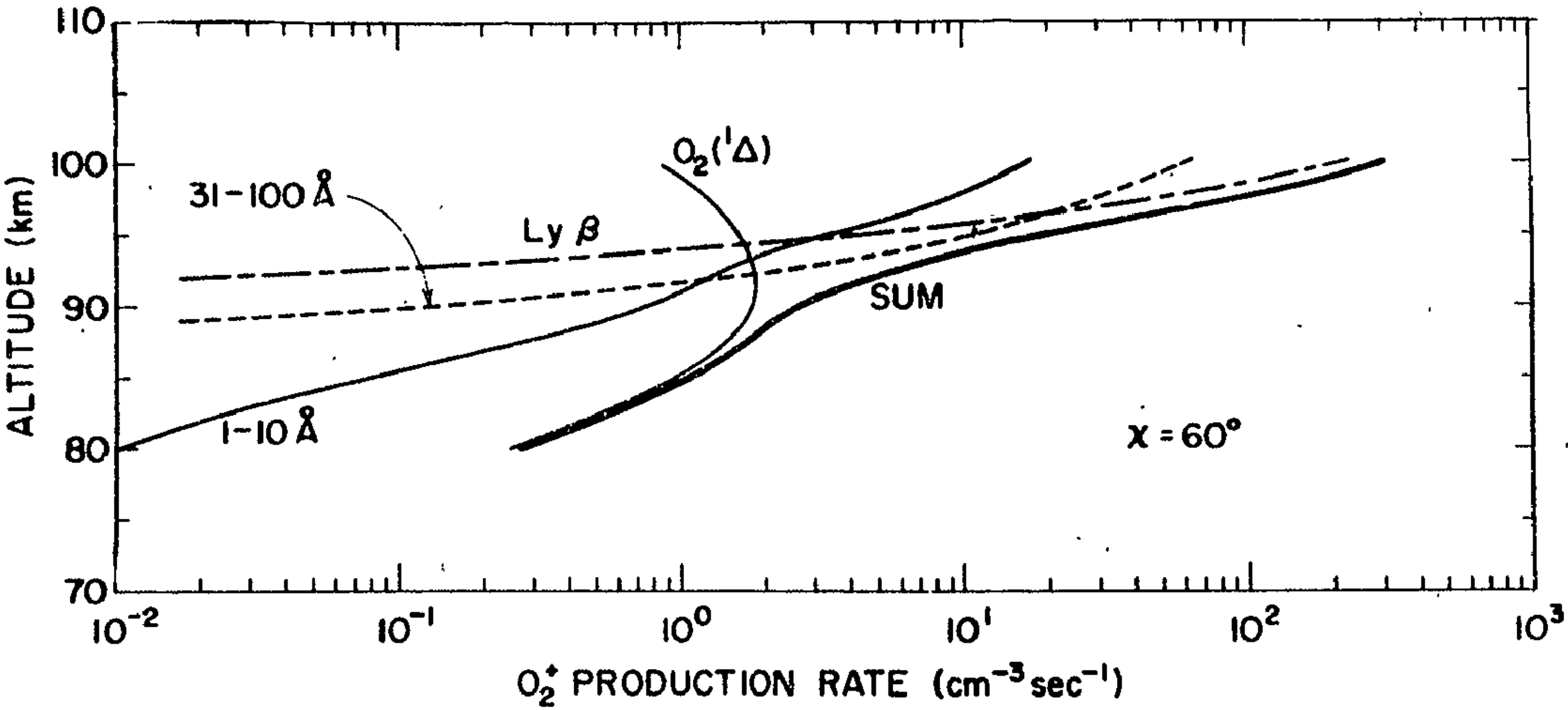


Figure 2.7 O_2^+ production rates for quiet-sun conditions and a solar zenith angle of 60° (SECHRIST, 1977).

2.4.2 Ion production. Figures 2.3, 2.4 and 2.5 show the major positive ions in the D region during a solar eclipse. The ions have been identified as $19^+(H_3O^+)$, $24^+(Mg)^+$, $30^+(NO^+)$, $32^+(O_2^+)$, $37^+(H_5O_2^+)$, $48^+(NO^+H_2O)$, $55^+(H_7O_3^+)$ and $56^+(Fe^+)$. Note the large concentrations of the long-lived metallic ions, Fe^+ and Mg^+. They may reach peak concentrations comparable with the electron concentration around 95 km.

From Figure 2.6 we see that NO^+ is the major positive ion produced in the altitude range 70-86 km. However, it is not the major ion observed in the mass spectrometer measurements (Figures 2.3-2.5). What is required is a fast process which can convert NO^+ to water cluster ions in this region. Evidence for such a fast conversion process is indicated in the eclipse measurements by the rapid disappearance and decay of NO^+ in the 78-86 km region while the water cluster ions actually increase between 83 and 90 km at totality. The speed of this unknown process may be appreciated by comparing Figures 2.4 and 2.5. The time difference between these two measurements is about 12.5 min in which time NO^+ at 84 km decays from about 500 cm^{-3} to about 20 cm^{-3}.

Table 2.1, from SECHRIST (1977), gives a possible reaction sequence for the fast conversion of NO^+ to hydrated protons. However, the $NO^+ + N_2 + M$ reaction is very difficult to measure in the laboratory because of the fast breakup to $NO^+(N_2)$ and its fast loss by switching with impurities such as NO, CO_2 or H_2O.

The behavior of negative ions in the D region is not very well understood. Negative ion reaction schemes have been proposed based on laboratory measurements of negative ion/molecule reactions, but there has been no substantial test of the schemes. Several negative ion studies based on mass-spectrometer measurements have been carried out, but they produce somewhat

Table 2.1 A possible reaction sequence for the fast conversion of NO^+ to hydrated protons (SECHRIST, 1977).

$$NO^+ + N_2 + M \rightarrow NO^+(N_2) + M,$$

$$NO^+(N_2) + CO_2 \rightarrow NO^+(CO_2) + N_2,$$

$$NO^+(CO_2) + H_2O \rightarrow NO^+(H_2O) + CO_2,$$

$$NO^+(H_2O) + N_2 + M \rightarrow NO^+(H_2O)(N_2) + M,$$

$$NO^+(H_2O)(N_2) + CO_2 \rightarrow NO^+(H_2O)(CO_2) + N_2,$$

$$NO^+(H_2O)(CO_2) + H_2O \rightarrow NO^+(H_2O)_2 + CO_2,$$

$$\vdots$$

$$\rightarrow NO^+(H_2O)_3 + CO_2$$

$$NO^+(H_2O)_3 + H_2O \rightarrow H^+(H_2O)_3 + HNO_2$$

conflicting or inconclusive results. The greatest difficulty is obtaining accurate in situ measurements. Figure 2.8 depicts negative ion results from in situ measurements made during the 1970 solar eclipse at Wallops Island. Heavy negative ions were found in largest concentrations below 92 km. Such large clusters are unexplainable by vapor phase reactions and may indicate that conglomerates are playing an important role in the negative ion chemistry.

From the large number of negative-ion reactions studied in the laboratory, O_2^- and O- are seen to be the primary ions. The major primary ion O_2^- is formed in a three-body attachment process.

$$O_2 + e + M \rightarrow O_2^- + M \tag{2.11}$$

The O_2^- ions in the lower D-region, where the $[O]/[O_3]$ ratio is small, can either charge transfer with O_3 or associate with O_2 to form O_4^-. O_3^- and O_4^- both react rapidly with CO_2 to form CO_3^- and CO_4^- ions. Reaction then takes place with NO to eventually form the stable, terminal ion NO_3^-, which hydrates to yield $NO_3^-(H_2O)_n$ ions. Large amounts of this hydrate can be seen in Figure 2.8.

Figure 2.8 also shows large count rates between 99 and 90 km for O^-, O_2^-, NO_2^- and NO_3^- (H_2O) on descent but not on ascent. NARCISI ET AL. (1972) state that such large concentrations have not been measured in this altitude range in seven other negative ion flights. The descent region was in darkness (97-97.5% solar obscuration) about 3.5 min longer than the ascent region (99.2% solar obscuration). However, there are no known processes by which large amounts of such ions could be produced at the fast rate required to balance the known large loss rates due to ion-ion mutual neutralization reactions and associative detachment reactions with atomic

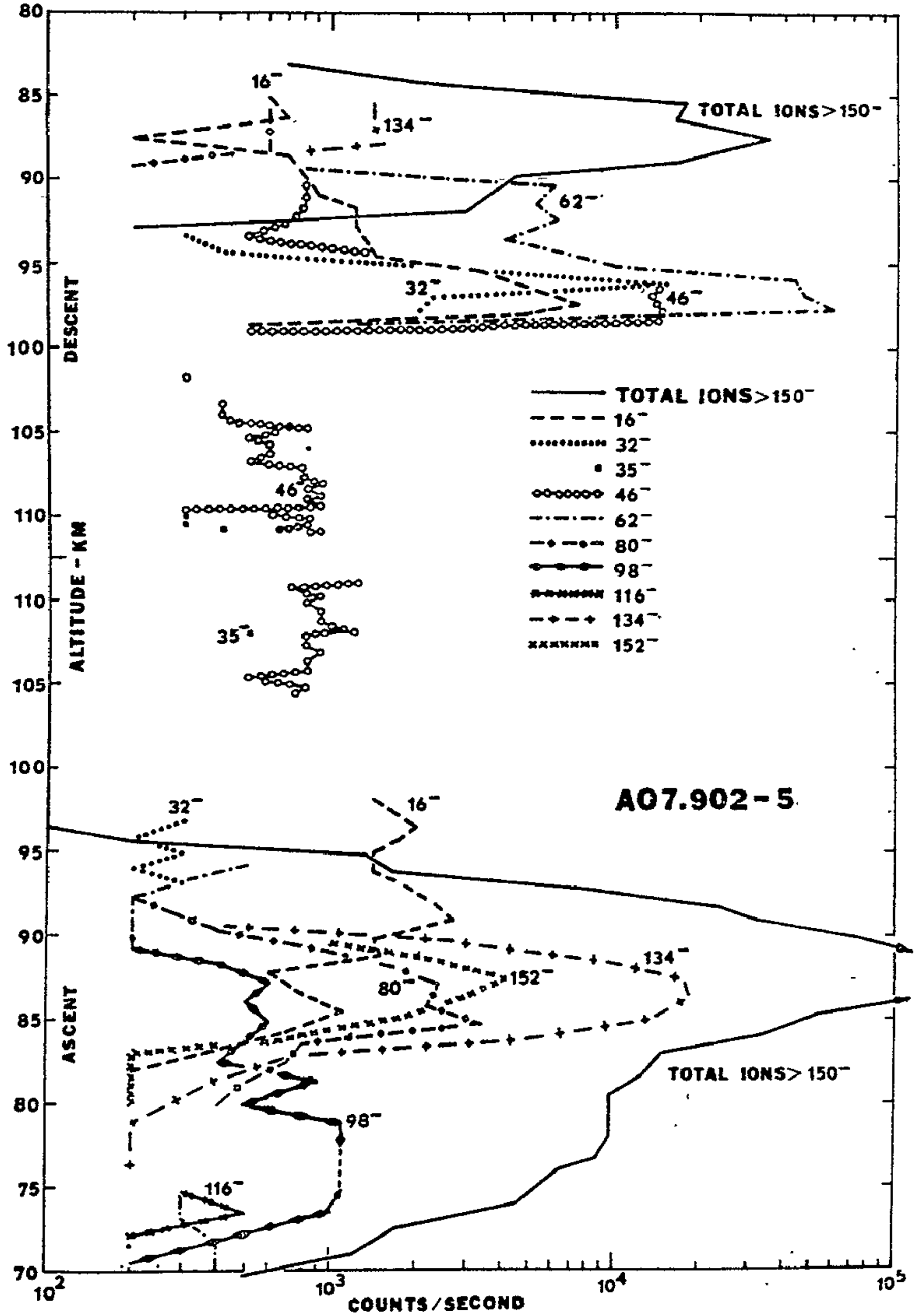


Figure 2.8 Negative-ion composition measurements shortly following totality. The solar obscuration was 99.1-99.2% in the 70-111 km upleg region and 97.8-96.6% in the 111-82 km downleg region (NARCISI ET AL., 1972).

oxygen (e.g., the reaction $O_2^+ + O \rightarrow O_3 + e$, is very fast).

2.4.3. Negative-ion ratio and effective recombination coefficient. To derive the effective recombination coefficient for the D region assume a simple positive molecular ion with concentration N_+ is created through photoionization at a rate q. Once created, the positive ions remain until dissociative recombination (coefficient α_d) with electrons or mutual neutralization with negative ions (coefficient α_i) occurs. Negative ions are formed through three-body collisional attachment at a rate of aN^2N_e where a is the rate coefficient, N is the neutral ion concentration and N_e is the electron concentration. Negative ions can be destroyed through photodetachment (coefficient ρ) or associative attachment through ionic reactions (coefficient δ) with minor neutral constituents (density N_m).

The continuity equations are:

$$\frac{dN_+}{dt} = q - \alpha_d N_+N_e - \alpha_i N_+N_- \tag{2.12}$$

$$\frac{dN_-}{dt} = aN^2N_e - N_-(\rho + \delta N_m) - \alpha_i N_+N_- \tag{2.13}$$

$$\frac{dN_e}{dt} = q - \alpha_d N_+N_e - aN^2N_e + (\rho + \delta N_m)N_- \tag{2.14}$$

Requiring the ionosphere to be electrically neutral implies

$$N_+ = N_- + N_e = (1 + \lambda) N_e \tag{2.15}$$

If we take $\frac{dN_-}{dt} = 0$ in 2.13, we have

$$\lambda = \frac{N_-}{N_e} = \frac{aN^2}{(\rho + \delta N_m + \alpha_i N_+)} \tag{2.16}$$

During the day the denominator of this expression is dominated by the photodetachment coefficient ρ. ρ varies little with height since the D region is optically thin to the radiation involved (visible and long ultra-violet). In contrast, aN^2 and δN_m decrease exponentially with height since the molecular concentrations N and N_m decrease with height. So, by day, λ decreases upward and is small above 90 km. At night, however $\rho \simeq 0$ and λ depends on the ratio $aN^2/\delta N_m$ (RISHBETH ET AL., 1969).

Figure 2.9, from RISHBETH ET AL. (1969) gives approximate values of λ in the D-region. The heights at which $\lambda \approx 1$ are 69 km in the daytime and 89 km at night.

Solving equations 2.12-2.15 for the time variation of electron density:

$$\frac{dN_e}{dt} = \frac{q}{1+\lambda} - (\alpha_d + \lambda\alpha_i)\, N_e^{\,2} - \frac{N_e}{1+\lambda}\,\frac{d\lambda}{dt} \tag{2.17}$$

Assuming that $\frac{d\lambda}{dt} \simeq 0$ (valid above 75 km), equation 2.17 becomes

$$\frac{dN_e}{dt} = \frac{q}{1+\lambda} - (\alpha_d + \lambda\alpha_i)N_e^{\,2} \tag{2.18}$$

so that the steady-state electron density is

$$N_e^{\,2} = \frac{q}{(1+\lambda)(\alpha_d + \lambda\alpha_i)} \tag{2.19}$$

The effective recombination coefficient, α_e, is therefore

$$\alpha_e = (1+\lambda)(\alpha_d + \lambda\alpha_i) \tag{2.20}$$

Electron density variations in the D region during a solar eclipse are illustrated in Figure 2.10 (MECHTLY ET AL., 1972). Electron densities below 85 km decayed markedly during totality. Mechtly et al. characterized the loss of electrons during totality by a recombination-like loss coefficient, A, and by an attachment-like loss coefficient, B. Calculated from electron density alone, A is roughly $1 \times 10^{-4} cm^3 sec^{-1}$, and B is practically constant at $8 \times 10^{-3} sec^{-1}$ below 87 km. Because the recombination coefficient A is

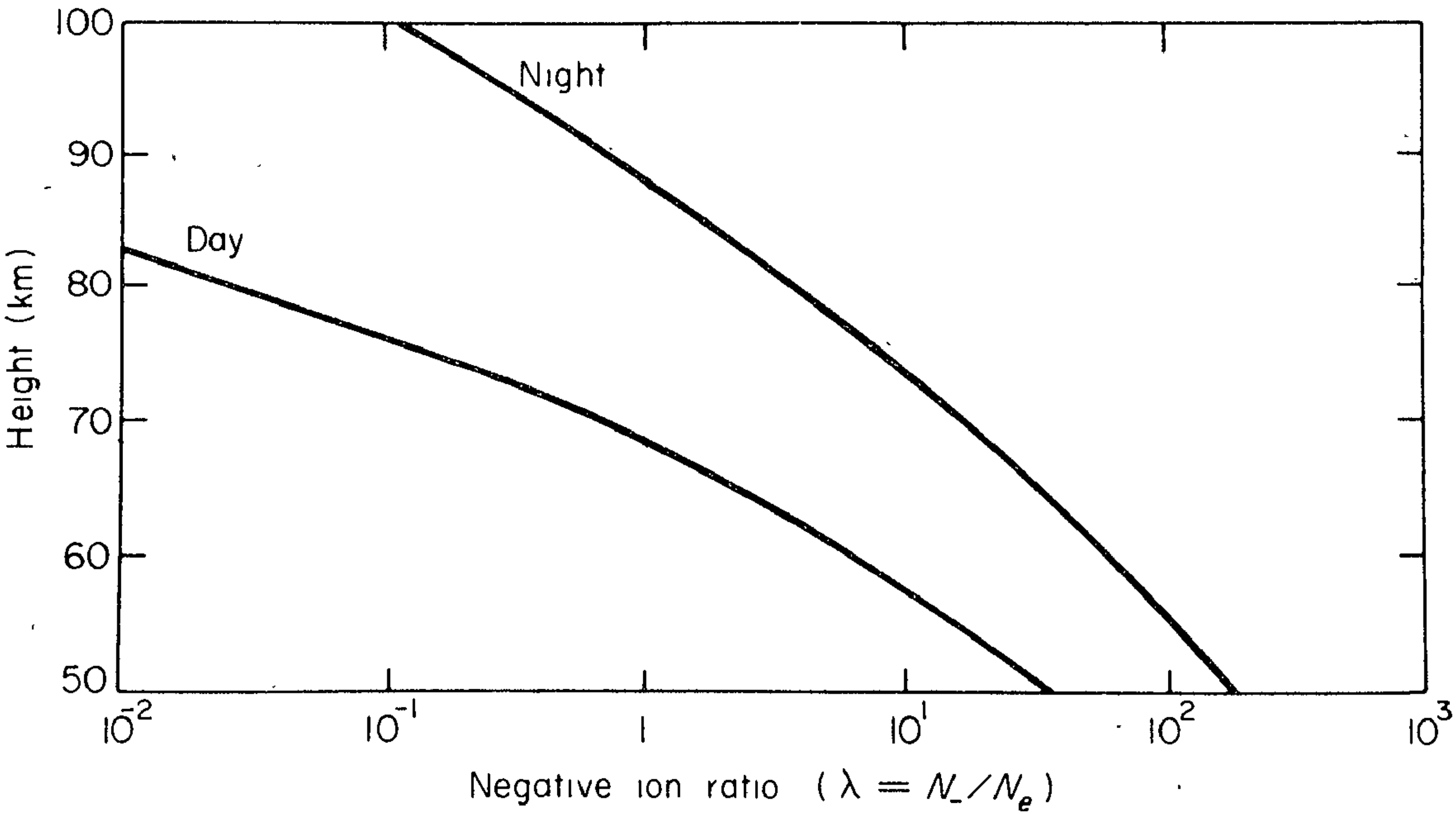


Figure 2.9 Approximate values of the negative-ion ratio in the normal D region [based on data given by Nicolet and Aiken (1960) and Aiken (1962)] (RISHBETH ET AL., 1969).

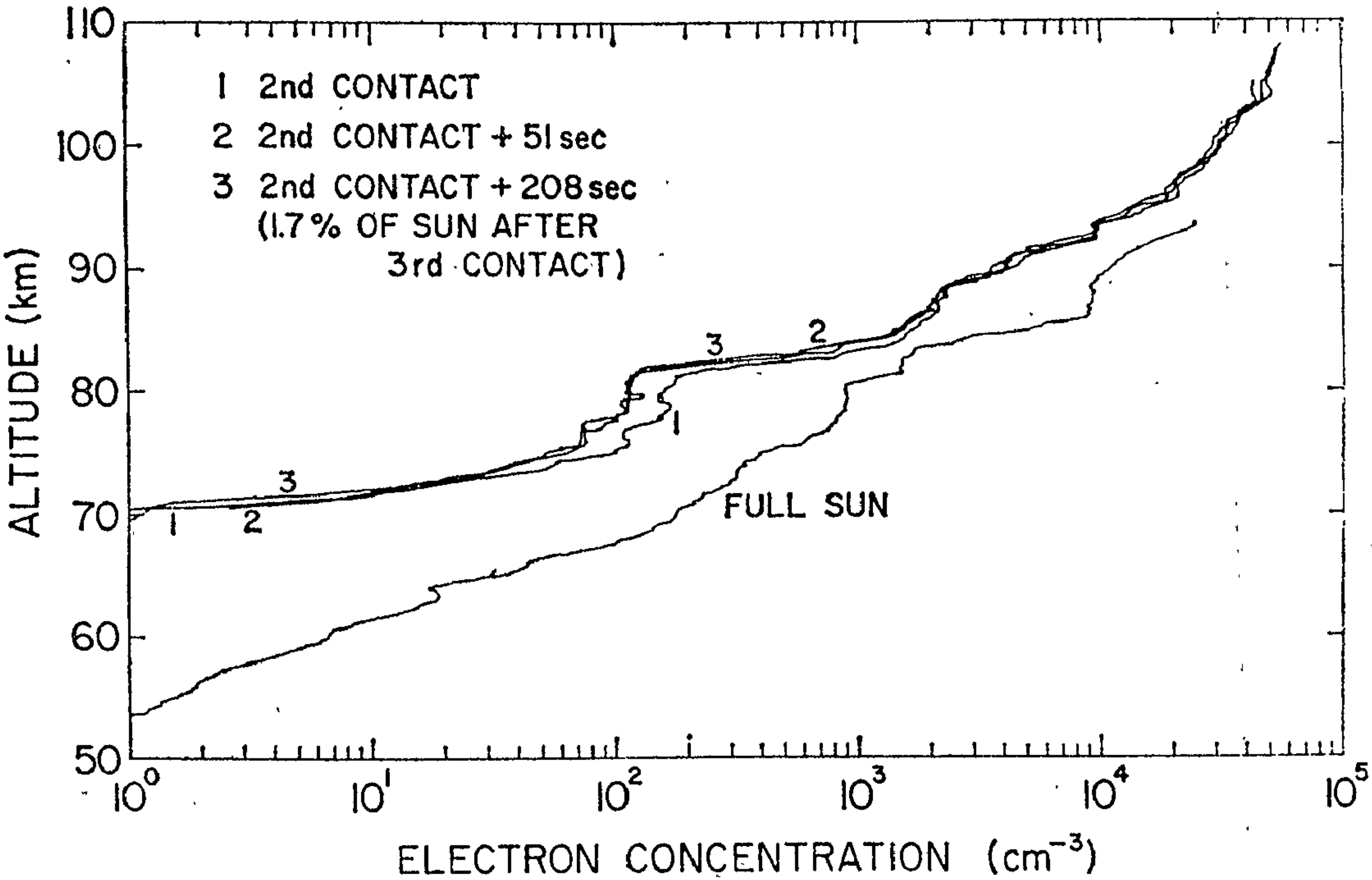


Figure 2.10 Rocket profiles of electron concentration from the solar eclipse of 7 March 1970 at Wallops Island (MECHTLY ET AL., 1972).

too large by a factor of 10, it is likely that negative ions will have to be considered in calculating α_e (SECHRIST, 1977). Unfortunately, the value of λ is extremely uncertain. Estimates of the daytime negative ion ratio range from 10 between 70 and 80 km (SECHRIST, 1977) to 1 at 49 km (HULTQVIST, 1963).

3. ELECTRON PRECIPITATION EFFECTS IN THE LOWER IONOSPHERE

3.1 Introduction

The precipitation of energetic electrons is responsible for many effects in the ionosphere, the most visible one being the aurora. Figure 3.1 (from VOSS AND SMITH, 1977) illustrates the complexity of electron precipitation.

The electrons are produced by a solar eruption and are then carried outward by the solar wind. At the point where the solar wind interacts with the earth's magnetic field (magnetopause) some particles become trapped and then spiral down the field lines to the lower ionosphere. The mechanisms by which the particles become trapped and are then accelerated are unknown.

The electrons travel down the field lines to collide with such atmospheric constituents as N_2, O_2 and O: leading to ionization, excitation and dissociation. The N_2^+ then emits radiation in several bands. When N_2^+ recombines some of the molecules are in excited states and also emit several bands.

Each ionization process by a high-energy electron produces a low-energy electron, called a secondary electron. The secondary electron often has sufficient energy to ionize or excite other atoms and molecules. The most common emission, 557.7 nm (greenish-yellow), is initiated in part by the excitation of O by secondary electrons.

Oxygen molecules are also excited by incoming electrons. The first negative band (580 to 680 nm) of O_2^+ is partly responsible for the crimson-red color. When O_2^+ recombines it usually dissociates, leaving one of the dissociated atoms in an excited state. After emitting radiation at 557.7 nm

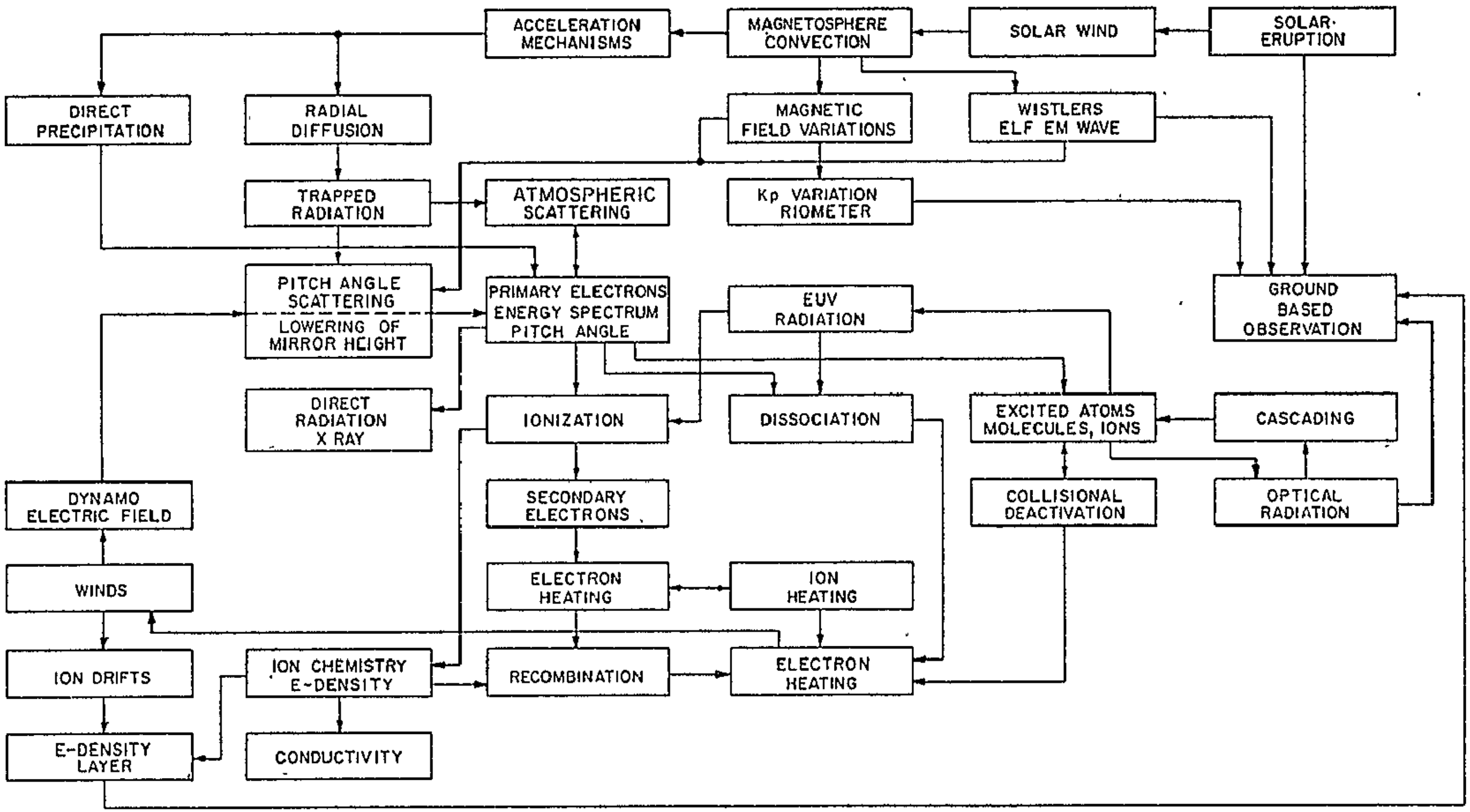


Figure 3.1 This figure illustrates the complex process involved with particle precipitation. Energetic particles are ejected from the sun with kinetic energies of a few hundred electron volts and travel as a solar wind to the earth's magnetosphere. Many mechanisms are associated with transport, acceleration, and precipitation of these particles. A complex series of atmospheric reactions may occur such as emission of radiation, ionization, composition changes, changes of temperature and generation of winds (VOSS AND SMITH, 1977).

the excited oxygen atom must emit radiation at 630 nm before returning to the ground state. This emission (dark-red color) can be seen in the upper part of a curtain-like form when it is very intense.

As the initial particle energy is reduced to about 7 eV, the predominant energy loss mechanism is vibrational excitation of molecular nitrogen. The last few electron volts are lost to ambient electrons by elastic collisions and, hence, increase T_e (ambient electron temperature).

Ionization rates versus altitude for different energy intervals are shown in Figure 3.2 (WHITTEN AND POPPOFF, 1965). These curves also represent the penetration depth of the electrons. Electrons in the 10-20 keV range penetrate to 95 km, whereas electrons near 200 keV reach nearly 72 km.

Although high-energy particles are needed to ionize the D region directly, bremsstrahlung radiation, generated by low-energy electrons, can penetrate to the lower altitudes.

Figures 3.3 and 3.4 are examples of bremsstrahlung contributions to ionization in the D region. Figure 3.3 gives the peak bremsstrahlung ionization rate profile for electrons with energies between 15 and 55 keV and with energies between 15 and 195 keV. Figure 3.4 shows the combined peak ionization rate profile for electrons plus bremsstrahlung. Although 55 keV electrons only penetrate to nearly 85 km, with a peak ionization rate of about 10^5 cm^{-3} sec^{-1}, bremsstrahlung can contribute significantly to the ionization rate below 85 km.

3.2 Ionospheric Emissions

3.2.1 Emission mechanisms. Energetic electrons can excite atmospheric species in several ways (WHITTEN AND POPPOFF, 1971).

Inelastic Collisions: Fast electrons can excite atoms or molecules by

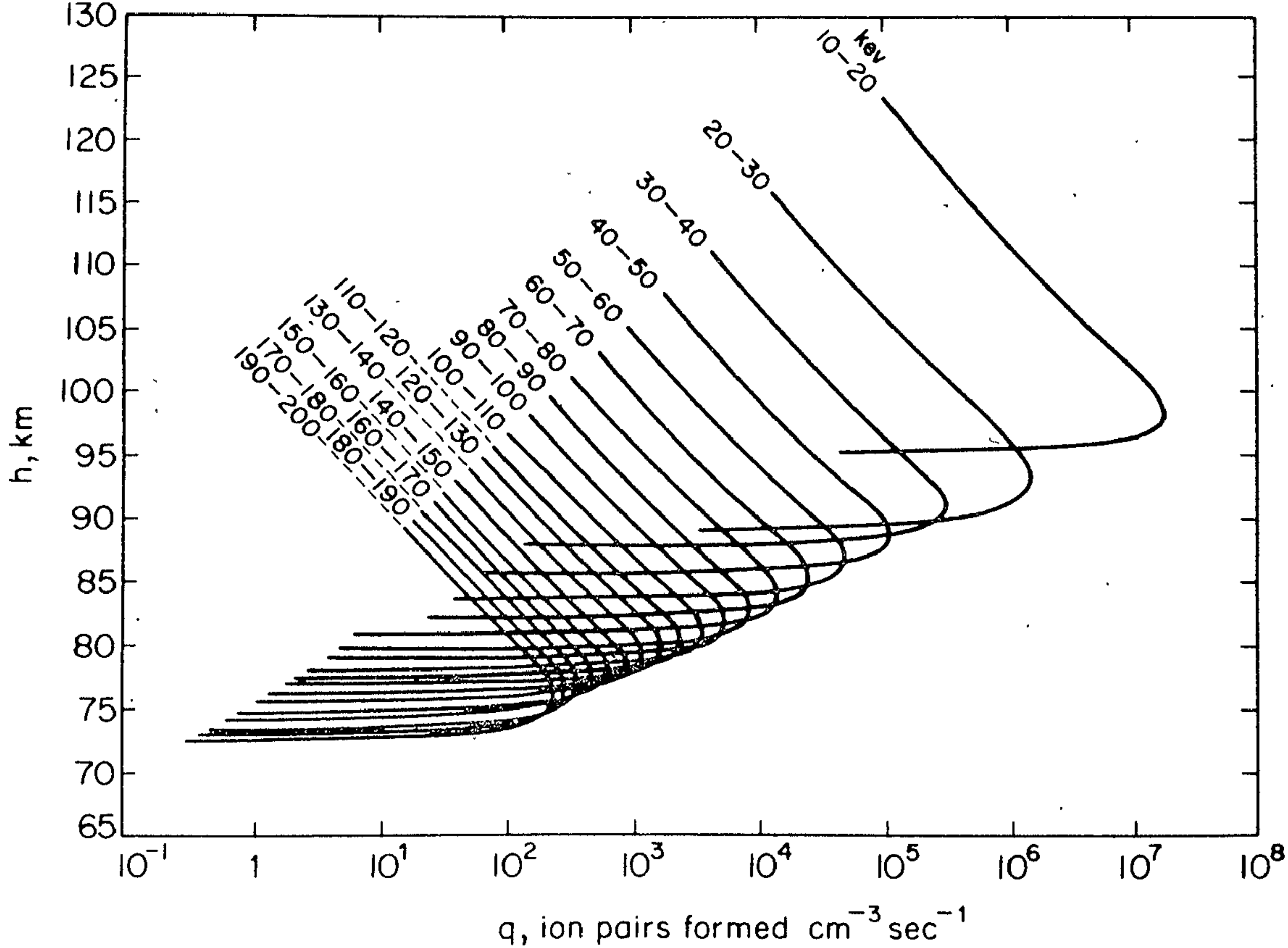


Figure 3.2 Ionization rate versus altitude (per energy interval) (WHITTEN AND POPPOFF, 1965).

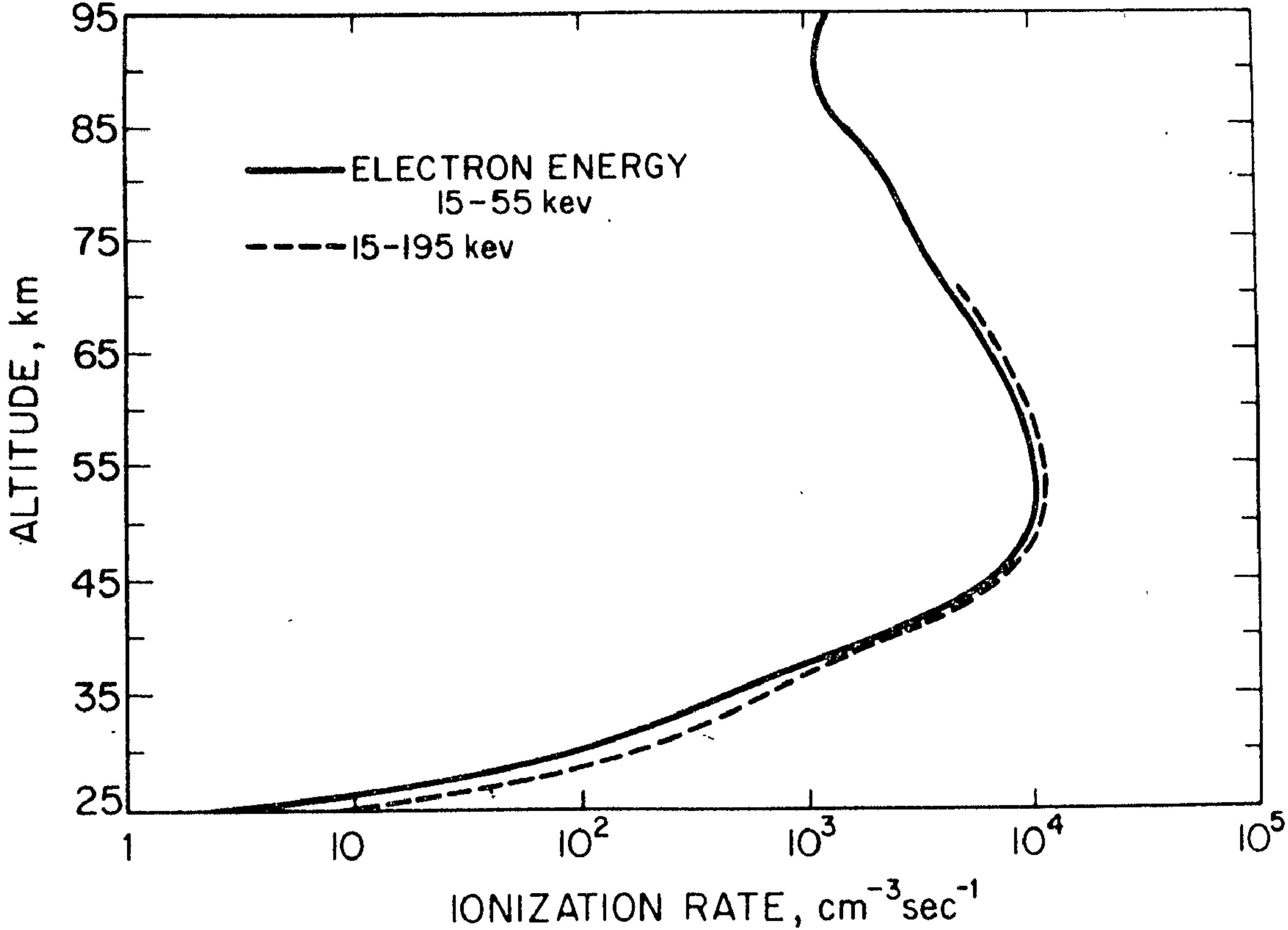


Figure 3.3 Peak bremsstrahlung ionization rate profile (WHITTEN AND POPPOFF, 1965).

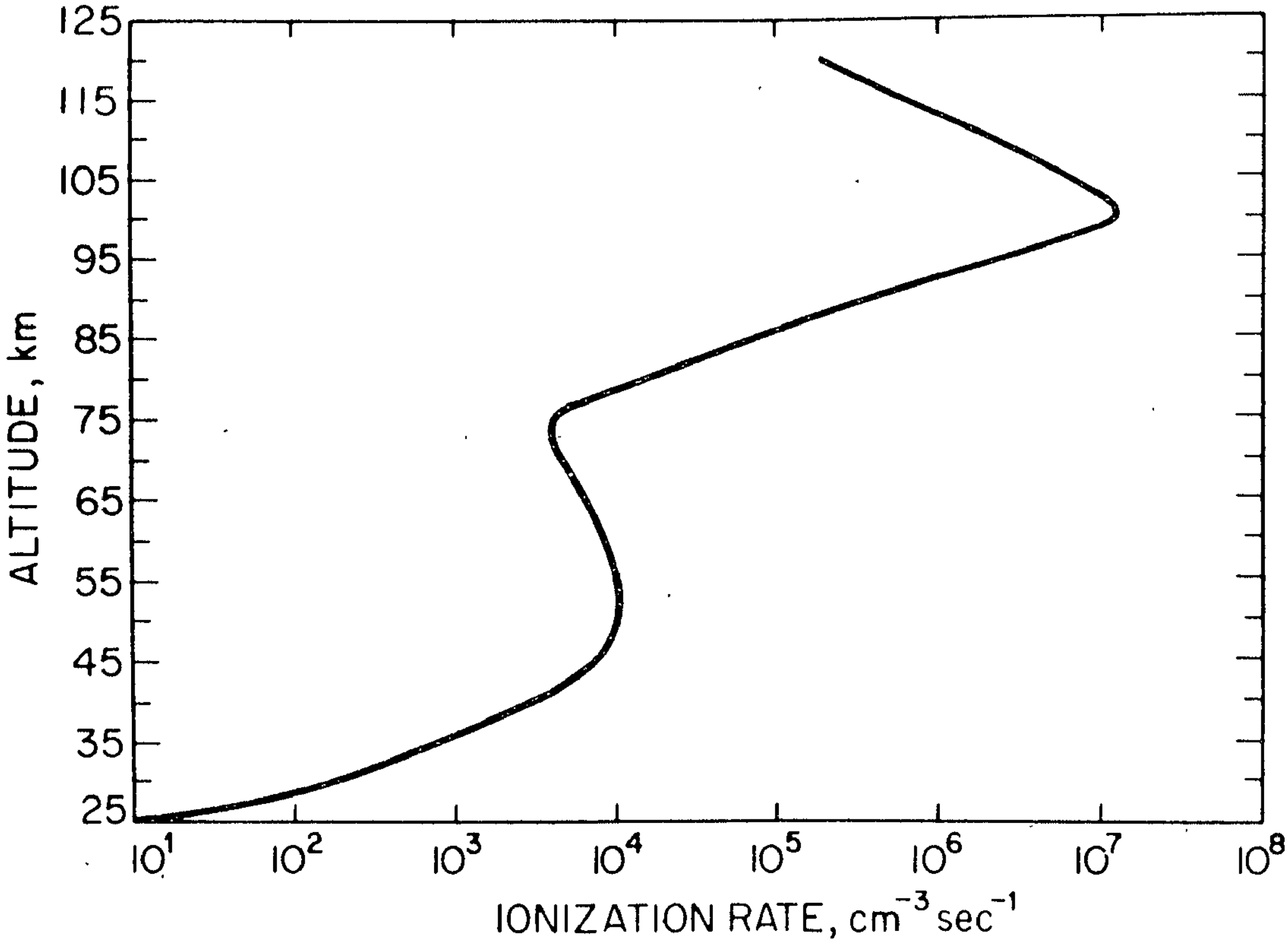


Figure 3.4 Peak ionization rate profile -- electrons plus bremsstrahlung (WHITTEN AND POPPOFF, 1965).

giving up a small amount of energy on impact and exciting the atoms or molecules to a higher energy state, or by dissociating molecules and leaving one or more products excited, or by simultaneously exciting and ionizing molecules or by simultaneously dissociating, ionizing, and exciting.

$$X + e \rightarrow X^{*} + e \tag{3.1}$$

$$X_2 + e \rightarrow X + X^{*} + e \tag{3.2}$$

$$X_2 + e \rightarrow X_2^{+*} + 2e \tag{3.3}$$

$$X_2 + e \rightarrow X + X^{+*} + 2e \tag{3.4}$$

Recombination: The recombination of thermal ions and thermal electrons results in excess energy which may be dissipated in kinetic energy or in the excitation of one or more of the products.

$$X^{+} + e \rightarrow X^{*} \tag{3.5}$$

$$X_2^{+} + e \rightarrow X^{*} + X^{**} \tag{3.6}$$

3.2.2 Emission rate. The basic unit of energy emission from auroras is the Rayleigh. Auroras have an apparent surface brightness, but, in fact, the emission is from a column of excited gas along the line of sight of the observer. Assuming no absorption or energy transfer complications within the emitting column, the output is determined by

$$4\pi I = \int_0^{\infty} V(r)\, dr \tag{3.7}$$

where $V(r)$ (photons cm^{-3} sec^{-1}) is the volumetric emission rate at a distance r from the observer. If I is in units of 10^6 photons cm^{-2} (column) sec^{-1} $ster^{-1}$, a Rayleigh (R) is defined as $4\pi I$ or the apparent emission rate of 10^6 photons cm^{-1} (column) sec^{-1}.

Auroras are also classed according to an International Brightness Coefficient (IBC) scale. On this scale IBC I = 1 kR; IBC II = 10 kR; IBC III = 100 kR; and IBC IV = 1000 kR.

3.2.3 Auroral spectra. Auroral spectra in the lower ionosphere are produced mainly by excited nitrogen and oxygen, neutral and ionized, molecular and atomic. The more prominent emissions are given in Table 3.1. Characteristics and production mechanisms of these emissions are summarized below.

Molecular Nitrogen: The N_2 band systems are most likely excited by the lower energy secondary electrons. It is possible for excitation of higher levels to occur in steps, i.e., excitation to a lower level by one collision followed by excitation to a higher level by a second collision.

Three reactions capable of producing the excited levels are:

$$e + N_2 (X^1\Sigma_g^+) \rightarrow e + N_2(A^3 \Sigma_u^+) \quad (3.8)$$

$$\rightarrow e + N_2(B^3\Pi_g) \quad (3.9)$$

$$\rightarrow e + N_2(C^3\Pi_u) \quad (3.10)$$

The excitation of N_2 bands may be an important factor in the slowing down of electrons in the 10-15 eV region. This effect may, in turn, influence the population of other species, such as $O(^1D)$, that are believed to be produced by collisions with slow electrons.

The N_2^+ excited state is probably produced simultaneously with ionization. The state that emits the Meinel band is produced by:

Table 3.1 Prominent auroral emissions.

SPECIES	TRANSITION	BAND OR LINE	INTENSITY[†]
N_2	$B^3\Pi_g \rightarrow A^3\Sigma_u^+$	first positive (red-infrared)	500-2400
	$C^3\Pi_u \rightarrow B^3\Pi_g$	second positive (violet-ultraviolet)	50-400
	$A^3\Sigma_u^+ \rightarrow X^1\Sigma_g^+$	Vegard-Kaplan (ultraviolet)	∿100
N_2^+	$A^2\Pi_u \rightarrow X^2\Sigma_g^+$	Meinel (infrared)	700-2000
O_2	$b^1\Sigma_g^+ \rightarrow X^3\Sigma_g^-$	atmospheric (red)	∿200
	$a^1\Delta_g \rightarrow X^3\Sigma_g^-$	infrared atmospheric	10^4-10^5
O	$^1D \rightarrow {}^1S$	557.7 nm (green line)	100*
N	$^2D \rightarrow {}^2P$	1040.0 nm (infrared)	100

Adapted from WHITTEN AND POPPOFF (1971) except †from OMHOLT (1971).

*All other intensities are normalized to this value for λ557.7.

$$e + N_2 (X^1 \Sigma_g^+) \rightarrow N_2^+ (A^2 \Pi_u) + 2e \tag{3.11}$$

Molecular Oxygen: Metastable states of molecular oxygen are produced by secondary electrons as follows:

$$e + O_2 (X^3 \Sigma_g^-) \rightarrow e + O_2 (b^1 \Sigma_g^+) \tag{3.12}$$

$$e + O_2 (X^3 \Sigma_g^-) \rightarrow e + O_2 (a^1 \Delta_g) \tag{3.13}$$

The $b^1 \Sigma_g^+$ state is also excited by thermal collisions with metastable oxygen atoms:

$$O(^1D) + O_2(X^3 \Sigma_g^-) \rightarrow O(^3P) + O_2 (b^1 \Sigma_g^+) \tag{3.14}$$

The 864.5 nm band of the atmospheric system emitted by the $b^1 \Sigma_g^+$ state is observed; the 761.9 nm band is absorbed by atmospheric oxygen. Similarly, 1.58 μm emission is observed for the infrared atmospheric system emitted by the $a^1 \Delta_g$ state; the 1.27 μm band is absorbed.

Atomic Nitrogen and Oxygen: Excited states of neutral nitrogen atoms are produced by electron impact and by dissociative recombination.

$$e + N_2 \rightarrow N + N^* + e \tag{3.15}$$

$$N_2^+ + e \rightarrow N + N^* \tag{3.16}$$

Similar reactions produce excited oxygen atoms except that electron impact on the atomic species is also possible. The metastable states can be produced as the result of impact with slow electrons with energies at or above the energies of the excited levels.

The metastable states of atomic oxygen can also be depopulated without the emission of radiation by collisions with nitrogen and/or oxygen molecules; this is one way for exciting the atmospheric system of molecular oxygen. The depopulation of metastable levels by collision is called "quenching".

3.3 Ionization by Energetic Electrons

3.3.1 Ion production rate. Methods of calculating the ion production rate for energetic particles may be classified into empirical, Monte Carlo and Boltzmann equation. Each method differs according to the manner in which scattering is treated. A brief outline of the empirical method will be given here. See VOSS AND SMITH (1977) for a more detailed treatment of the above three methods.

The empirical method uses a range-energy loss rate equation to compute the energy loss in each height increment. The ion-production rate q(z) is then obtained by dividing the energy loss rate by the average energy loss per collision (35 eV).

In calculating q(z), REES (1963) used the experimentally determined energy loss rate equation of GRUN (1957)

$$\frac{dE}{dR} = -AE^{-m} \text{ (E in keV)} \qquad (3.17)$$

where $A = 1.25 \times 10^4$ and $m = 0.75$ for electrons over the energy range 4 keV to 100 keV and R is given in units of $cm^2\ g^{-1}$. Equation 3.17 was determined empirically for electrons in air and takes into account the scattering effects.

Equation 3.17 is integrated over height increments ΔZ_i, to give

$$\Delta E_i^{1.75} = 1.25 \times 10^7\ \overline{\rho_i}\ (\cos\alpha)\ \Delta Z_i \qquad (3.18)$$

where E_i is the energy (keV), α is the pitch angle and $\overline{\rho_i}$ is the average atmospheric density in the ith height interval ($kg\text{-}m^{-3}$). The calculation is repeated for different energy spectrums and pitch angle distributions.

This method is very useful for quasi-isotropic electron distributions and quickly gives a reasonably accurate ionization rate profile.

The weakness of this method is that the detailed effect of atmospheric scattering in changing the pitch angle is not taken into account. Thus, it is not suitable for pitch-angle distributions which are sharply peaked and, to a lesser extent, peaked energy spectrums.

3.3.2 Electron concentration. Electron auroras are characterized by a large range of temporal and altitude conditions. For auroral bursts longer than minutes, the situation can be approximated by the quasi-equilibrium forms of the continuity equations (2.12-2.15). However, for microbursts the equilibrium forms are not applicable.

The buildup of electron concentration following the onset of an ionizing pulse can be approximated by Equation 2.18. If q(z) is taken as constant after onset ($t \geq 0$), Equation 2.18 can be integrated to give

$$N_e\ (z,t) = \left[\frac{q(z)}{(1+\lambda)(\alpha_d + \lambda\alpha_i)}\right]^{\frac{1}{2}} \tanh\left[\left(\frac{q(z)(\alpha_d + \lambda\alpha_i)}{1+\lambda}\right)^{\frac{1}{2}} t\right] \quad (3.19)$$

To simplify integration of this non-linear differential equation (Equation 2.18) N_e at t = 0 is taken to be 0.

Figure 3.5 (from BREKKE, 1975) shows E-region electron density variations during a sudden commencement electron precipitation event. The time between profiles is 30 sec. The sudden commencement began at approximately 2054 UT and the rapid buildup in electron density over the next 90 sec is evident. The increase in electron density at 110 km conforms very closely to Equation 3.19 with a q of 4.5×10^4 el/cm^3 -sec.

The decay of electron concentrations after the abrupt cessation of ionization can be predicted by using the approximation

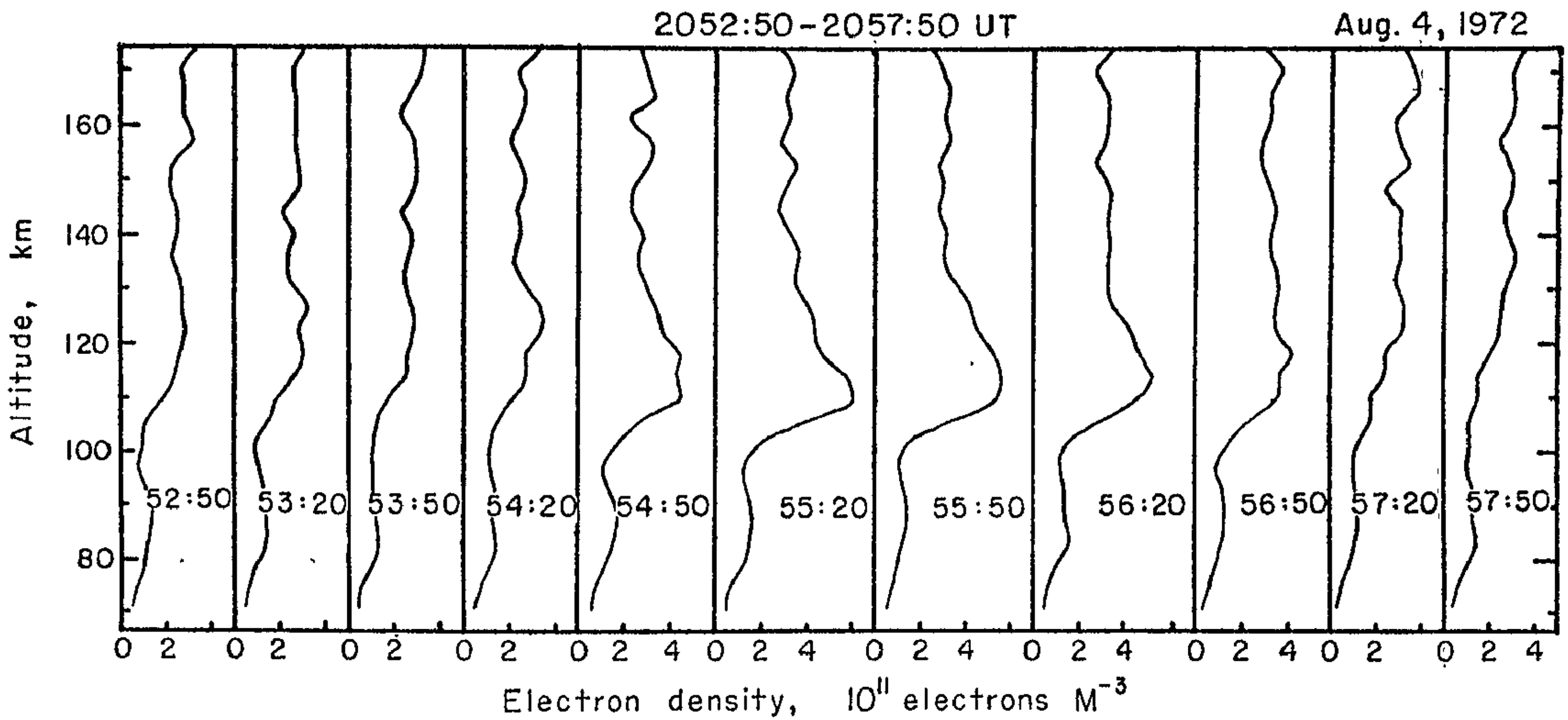


Figure 3.5 Electron density profiles observed every 30 sec, by the incoherent scatter radar at Chatanika, Alaska in the period of the sudden commencement between 2052:50 and 2057:50 UT on 4 August 1977 (BREKKE, 1975).

$$\frac{dN_e(z,t)}{dt} = -(\alpha_d - \lambda\alpha_i)\ N_e\ (z,t)^2 \qquad (3.20)$$

If we assume that λ is constant, but that $N_e = N_o$ when the pulse ceases, and integrate, we find that

$$N_e\ (z,t) = \frac{N_o(z)}{1 + (\alpha_d + \lambda\alpha_i)\ N_o(z)t} \qquad (3.21)$$

The electron density in Figure 3.5 does not decrease as rapidly as this equation predicts, but then the electron shower probably did not end abruptly.

3.3.3 $[NO^+]/[O_2^+]$ ratio. E-region positive ion composition during an IBC class II aurora is shown in Figure 3.6 (from NARCISI ET AL., 1974). Clearly, NO^+ and O_2^+ dominate over the entire E region with O^+ approaching 10% of the total ionization only at higher altitudes. Throughout the E region the $[NO^+]/[O_2^+]$ ratio is greater than one.

This would appear to be true for all auroral events. Figure 3.7 shows measured $[NO^+]/[O_2^+]$ ratios for eight auroral events (SWIDER ET AL., 1976). Curve seven is the $[NO^+]/[O_2^+]$ ratio for the ion measurements of Figure 3.6. There is no apparent correlation between these ratio profiles and the intensity of the aurora, or with any other auroral parameter, such as the K_p index. Curves 1-4 represent roughly IBC Class I events, curves 5-7 corresponding to Class II and curve 8 representing about an IBC Class III aurora.

D-region ion species between 84 and 72 km are plotted in Figure 3.8 (from NARCISI ET AL., 1974). There is a reversal to O_2^+ dominance between 82 and 86 km. This altitude range coincides with the region where a minimum in the NO concentration is found (MEIRA, 1971). Thus for smaller NO con-

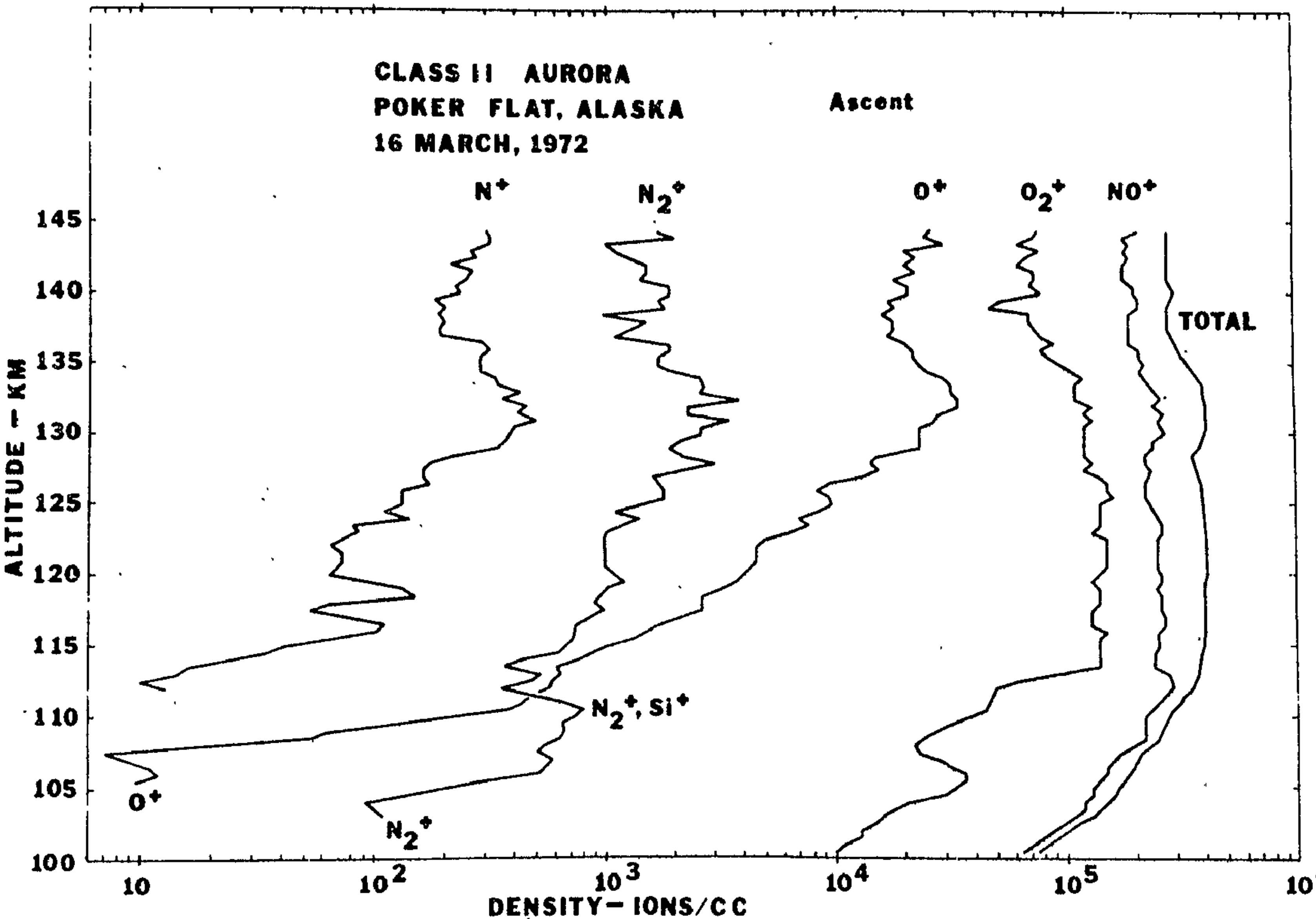


Figure 3.6 Auroral ion species concentrations in the E-region obtained on rocket ascent. There may be an admixture of Si^+ between 105 and 112 km in the N_2^+ profile (NARCISI ET AL., 1974).

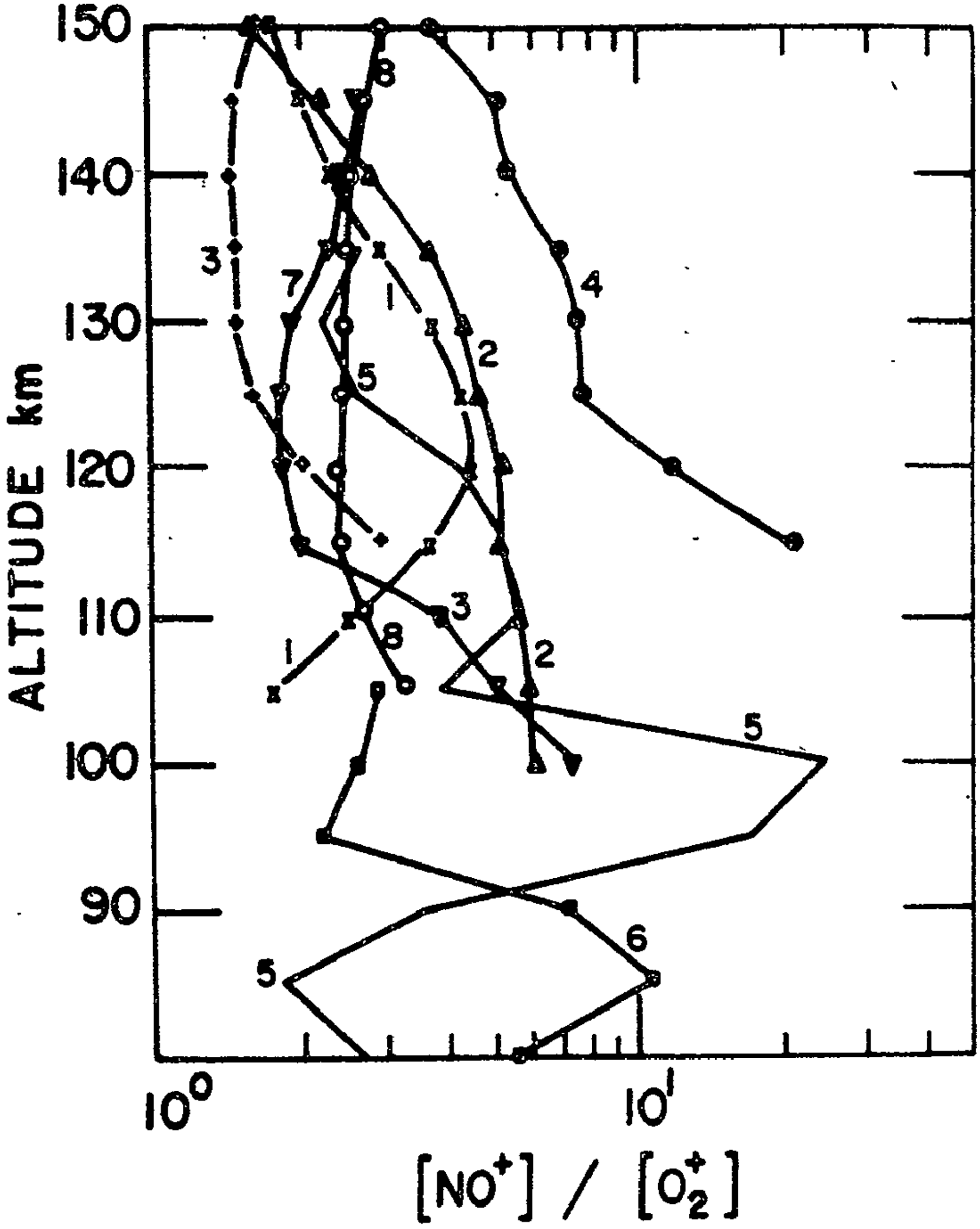


Figure 3.7 Measured NO^+/O_2^+ ionic concentration ratio for eight auroral flights (SWIDER AND NARCISI, 1977).

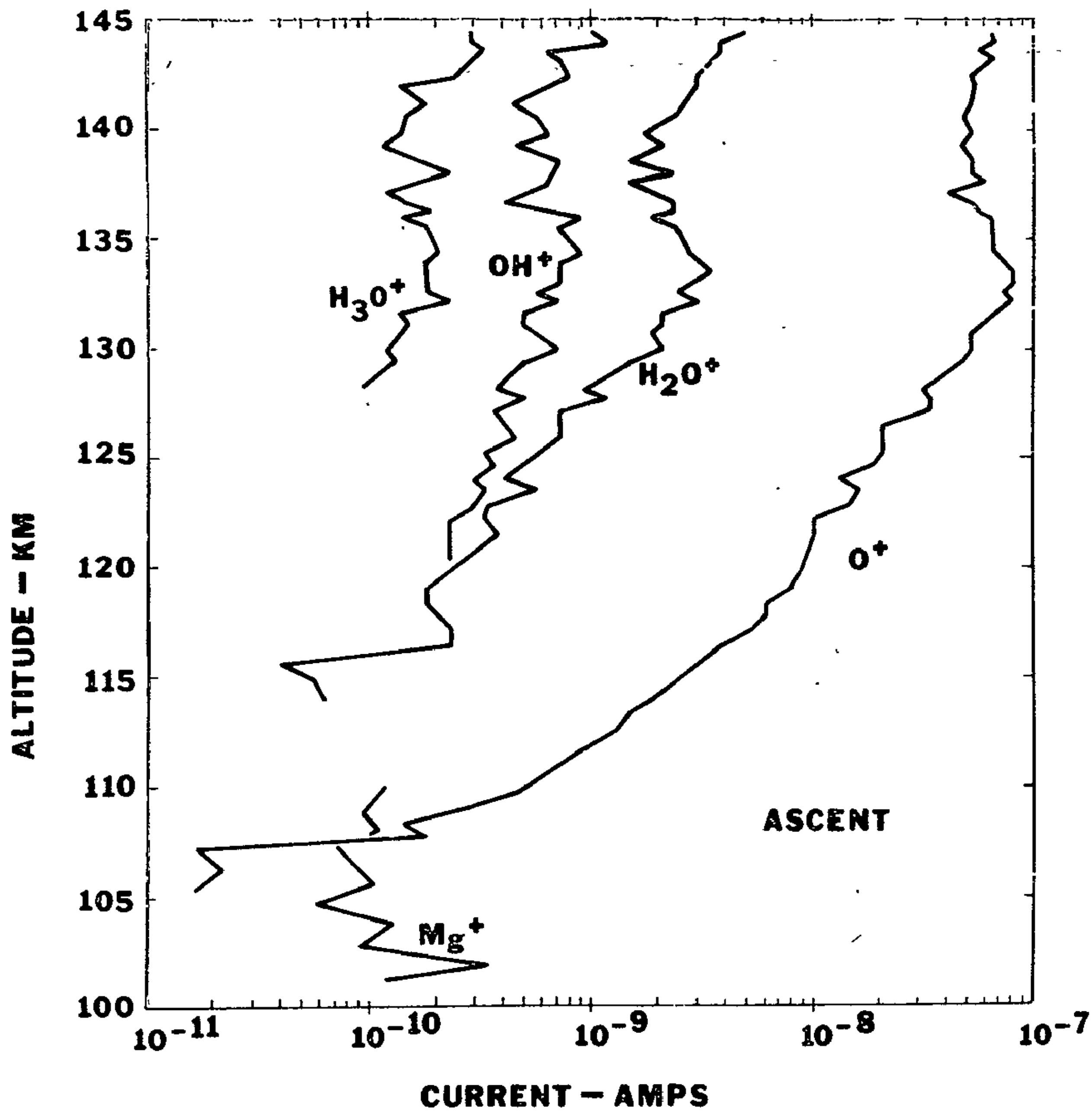


Figure 3.8 Current versus altitude profile showing the relative magnitudes of the H_2O^+, OH^+, H_3O^+ and O^+ ions during an IBC Class II aurora. The measured Mg^+ profile is also shown (NARCISI ET AL., 1974).

centrations, less O_2^+ is lost, and less NO^+ is produced by the reaction $O_2^+ + NO \rightarrow NO^+ + O_2$ leading to a decrease in the $[NO^+]/[O_2^+]$ ratio.

4. EXPERIMENTAL TECHNIQUES

4.1 Introduction

Three of the experiments aboard the eclipse payloads will be described in this chapter. They are:

(1) the Langmuir probe experiment - used to determine the relative electron density, electron temperatures, electron density fine structure and vehicle potential

(2) the propagation experiment - used to determine the absolute electron density for calibration of the Langmuir probe

(3) the energetic particle experiment - used to determine the fluxes due to precipitating particles.

Each of the following sections is devoted to one of the above experiments. The theory behind the experiment as well as the experimental implementation of the theory are discussed. The references consulted for each experiment are given at the end of each section.

Figure 4.1 illustrates the arrangement of the eclipse payloads. The section labeled Bern experiment is a mass spectrometer (positive or negative ions) operated by the University of Bern (Switzerland). The telemetry for the Bern experiment is located in the section immediately below the experiment. Also in this section is the NASA tone-ranging system (TRADAT) for trajectory determination. The clamshell nosecone on the top of each rocket is ejected just before entering the D-region to expose the spectrometer inlet.

The antennas and receivers for the radio propagation experiments (one operating at 2.225 MHz, the other at 5.040 MHz) are located in the section labeled "Illinois receiving". The remaining University of Illinois experiments are located in the Illinois experiments section together with their

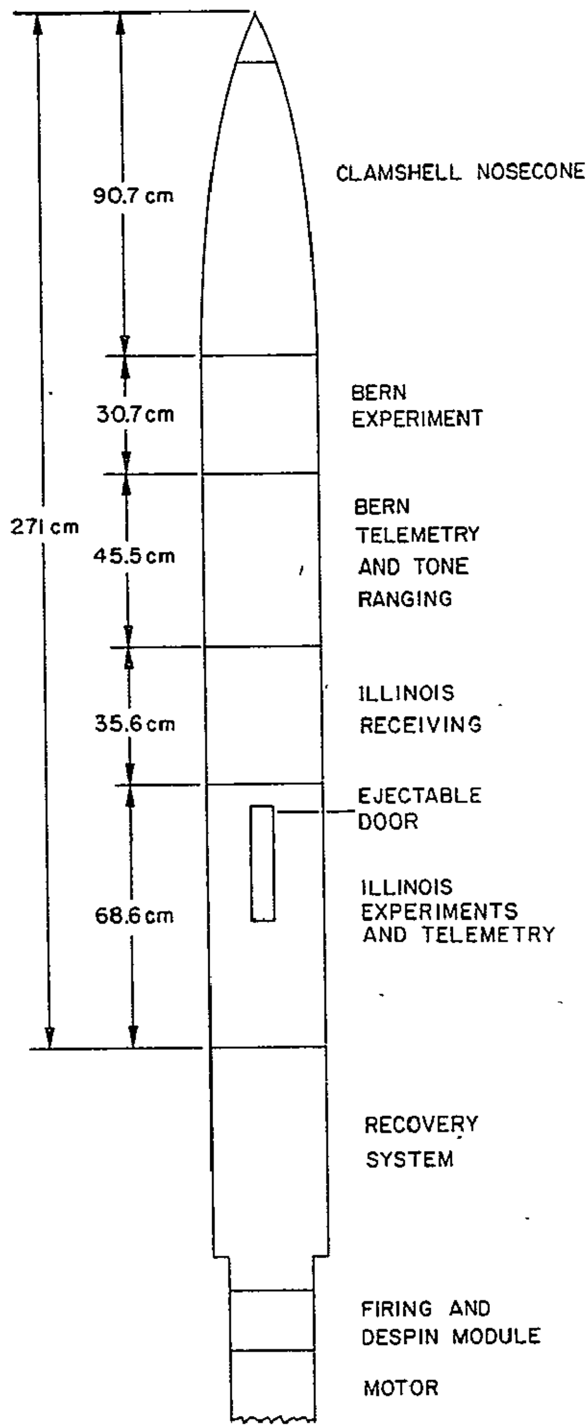


Figure 4.1 The general arrangement of the three payloads for the eclipse operation.

telemetry system. The ejectable doors (one on each side) open at approximately 60 km altitude to expose the detectors and probes.

At the rear of the payload are: the recovery system, containing a parachute and radio beacon; and the firing and despin module, containing the sensors and electronics for control of second stage ignition, payload/motor separation and rocket despin. (Despin reduces the rocket spin rate by increasing the moment of inertia. This is accomplished by releasing weights which are attached to the rocket by cables.)

Figure 4.2 is a sketch of the Illinois receiving, experiments and telemetry sections identifying the location of each experiment. The X-ray, visible light and Lyman-α experiments are discussed in BLISS AND SMITH (1980). A photograph of the fully assembled Illinois payload is reproduced in Figure 4.3.

The payload instruments are powered by a +30 V, 2 amp-hour battery pack. A regulated DC/DC converter module is used to obtain -30 V.

The mechanical timers are started at launch and control the ejection of the payload doors and the extension of the booms. (The booms are shown folded inside the payload in Figure 4.2 and are partially extended in Figure 4.3. The booms hold the two particle detectors and the two spherical Langmuir probes.) The door ejection circuit is armed at 21 km by a baroswitch. Baroswitches are also included to initiate calibrations for the Langmuir probe experiment and the Lyman-α experiment.

The spin magnetometer is used to determine the rocket spin rate as well as the rocket's precessional motion. The magnetometer outputs a voltage which is proportional to its alignment with the earth's magnetic field (e.g., a maximum when parallel to a magnetic field line; a minimum when anti-parallel to a magnetic field line).

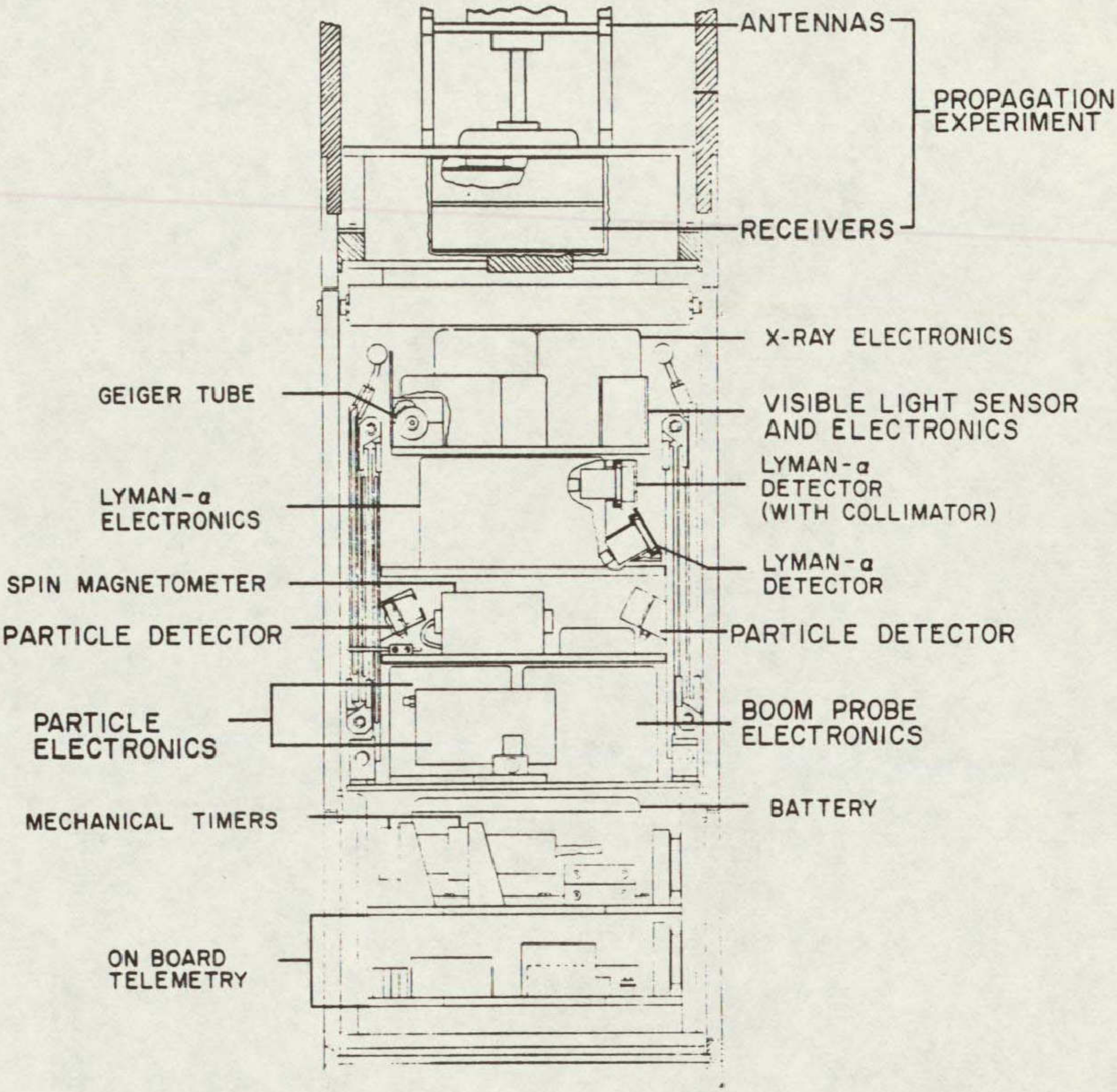


Figure 4.2 Arrangement of the University of Illinois experiments in the payload.

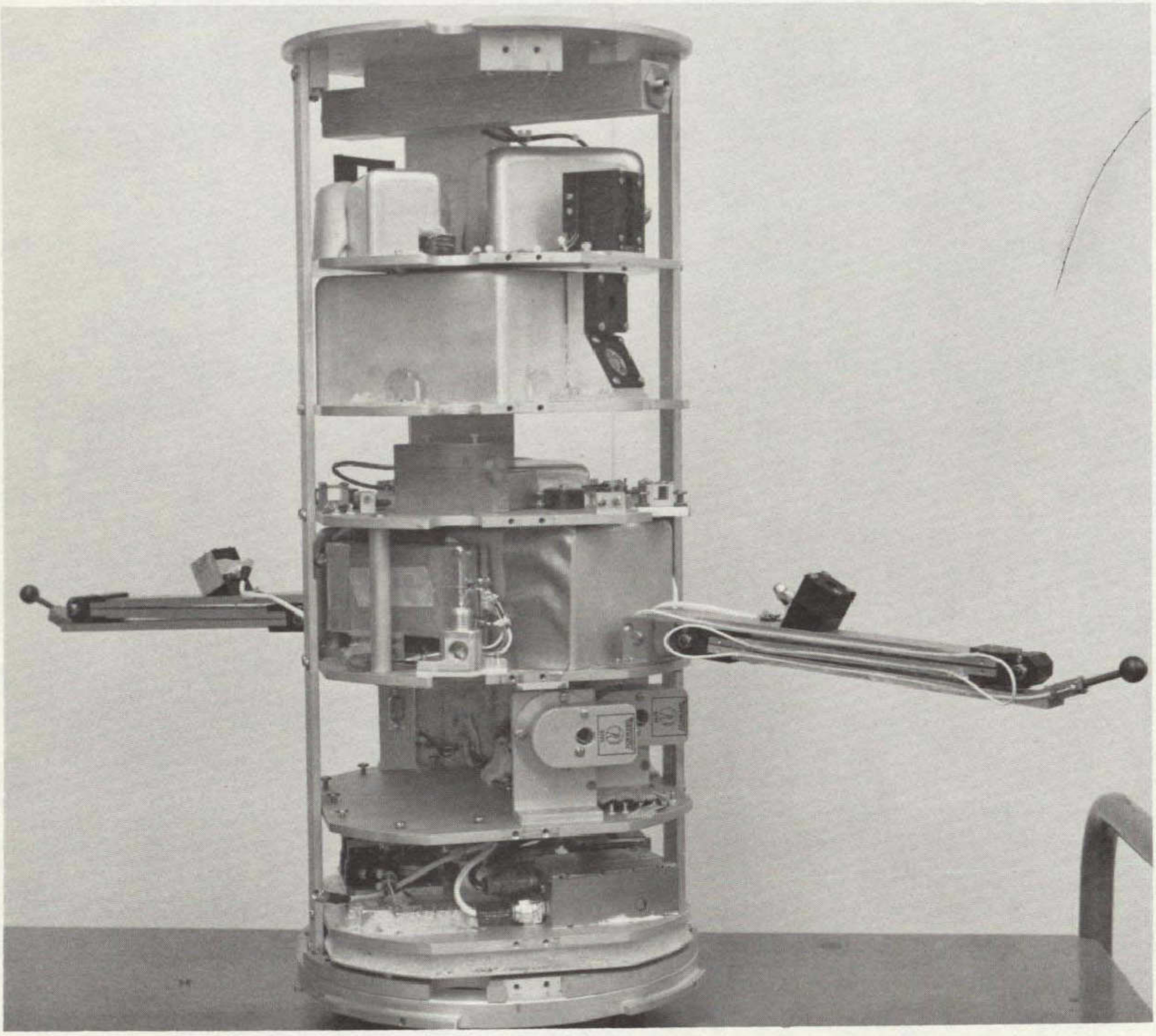


Figure 4.3 The lower section of one of the Nike Tomahawk payloads including the X-ray, Lyman-α, solar sensors and supporting instrumentation, the partially extended booms carrying the probes and the particle detectors.

The bottom deck of the payload contains the on-board telemetry system. The system is of the FM/FM type. Sixteen channels are frequency division multiplexed onto a single carrier frequency. The IRIG (Inter Range Instrumentation Group) proportional bandwidth system is used for channel assignment. Each channel uses a voltage controlled subcarrier oscillator (VCO) which will accept a 0 to 5 V input swing resulting in a ±7.5% deviation of the VCO center frequency. The outputs of the VCO's are mixed together and are then fed into an FM transmitter for transmission to the ground receiving station. Table 4.1 lists the channel assignments for the three eclipse payloads.

On the ground the transmitted signal is received with a high-gain tracking antenna and fed to an FM receiver. The receiver demodulated output is recorded, still in multiplexed form, on magnetic tape along with time code information and a 100 kHz reference signal. (The reference signal is used during playback to synchronize the digitizing process. This removes any digitization errors caused by tape speed variations.) Selected channels can be discriminated for real-time inspection. The two receiver modulation signals (channels 17 and 16) are immediately discriminated and sent to the UI van (see section 4.3.2) where they are used to close the loop for the radio propagation experiment. All of the real-time discriminated channels are recorded on a strip chart recorder along with time code information for immediate viewing of the data.

4.2 Langmuir Probe

Conceptually the probe is simple, consisting of a conducting electrode which is inserted into the plasma on an insulating support. The potential of the probe is varied with respect to the rocket body and the resulting current flow to the probe is telemetered to the ground. The resultant

Table 4.1 Nike Tomahawks 18.1020, 18.1021 and 18.1022 channel assignments.

Ch. 21	Geiger counter (0.1 - 0.8 nm X-rays)
20	Particle detector, output #1
19	Solar sensor
18	Boom probe, fine structure (2.5 V)
17	Receiver #1, modulation
16	Receiver #2, modulation
15	Boom probe, log scale
14	Particle detector, output #2
13	Particle detector, output #3
12	Lyman-α #1
11	Lyman-α #2
10	Spare
9	Spin magnetometer
8	Receiver #1, AGC
7	Receiver #2, AGC
6	Commutator (recovery system and door release monitor)

current versus voltage curve, or I-V characteristic, depends mainly on the ion and electron temperatures and the electron concentration.

Probe theory, on the other hand, is complicated because the probe surface is a boundary to the ambient plasma, and near boundaries the equations that govern plasma behavior change. There is no general theory for the Langmuir probe response under arbitrary plasma conditions, and many of the experimental problems need to be addressed.

4.2.1 Theory. Conventional Langmuir probe theory is valid between approximately 100 and 800 km altitude. Over this height range the probe is in the collisionless thin sheath mode of operation.

From the kinetic theory of gases, the current density, j_o, due to the random thermal motion of electrons in the plasma is

$$j_o = ne\bar{v}/4 \tag{4.1}$$

where n is the electron density and e the electron charge. $\bar{v}$ is the average velocity of the electrons and is given by

$$\bar{v} = (8kT_e/\pi m)^{1/2} \tag{4.2}$$

where k is the Boltzmann constant and m the electron mass. T_e is the electron temperature and is representative of the Maxwellian velocity distribution of the electrons. Electrons in the ionosphere are found to be in equilibrium among themselves (but not necessarily in equilibrium with the ions).

Equation 4.1 gives the current density to a probe which is at plasma potential. (At plasma potential ions and electrons adjacent to the probe are neither attracted to nor repelled from the probe surface.) When the probe is made negative with respect to the plasma potential, the electron current density is given by

$$j = j_o \exp(eV/kT_e) \quad V < 0 \tag{4.3}$$

This equation is valid for a probe of any geometry. For accelerating potentials, however, the variation of current with voltage does depend on the probe geometry. For a small sphere

$$j = j_o \left(1 + \frac{eV}{kT_e}\right) \qquad V > 0 \tag{4.4}$$

Figure 4.4 summarizes the I-V characteristic of a small spherical probe. The current into the probe is a combination of positive ions and electron components. The positive ion current is represented by the above equations but with the sign of potential (V) reversed.

4.2.2 Implementation. A block diagram of the Langmuir probe experiment is shown in Figure 4.5. The probe is used in two modes: fixed voltage and swept voltage. During the fixed voltage mode the mode relay is OFF and the probe is held at 4.05 V, with respect to the rocket body. During the swept voltage mode the probe voltage is swept from -1.35 V to 4.05 V in 0.5 sec. The mode relay is in the ON position during this time. The fixed voltage mode is used for measuring electron concentration and electron concentration fine structure, while the swept potential mode is used for determining electron temperature and vehicle potential.

Using Equations 4.3 and 4.4 with Figure 4.4 one can see that the probe I-V characteristic is exponential only between floating potential and plasma potential. It is linear otherwise. This indicates a way of determining electron temperature. If the probe current is differentiated with respect to voltage, the two linear portions of the I-V curve will yield a constant current versus voltage, while the exponential portion of the I-V curve will remain exponential. Further, if the logarithm is then taken a curve similar to Figure 4.6 will result. Electron temperature is then inversely pro-

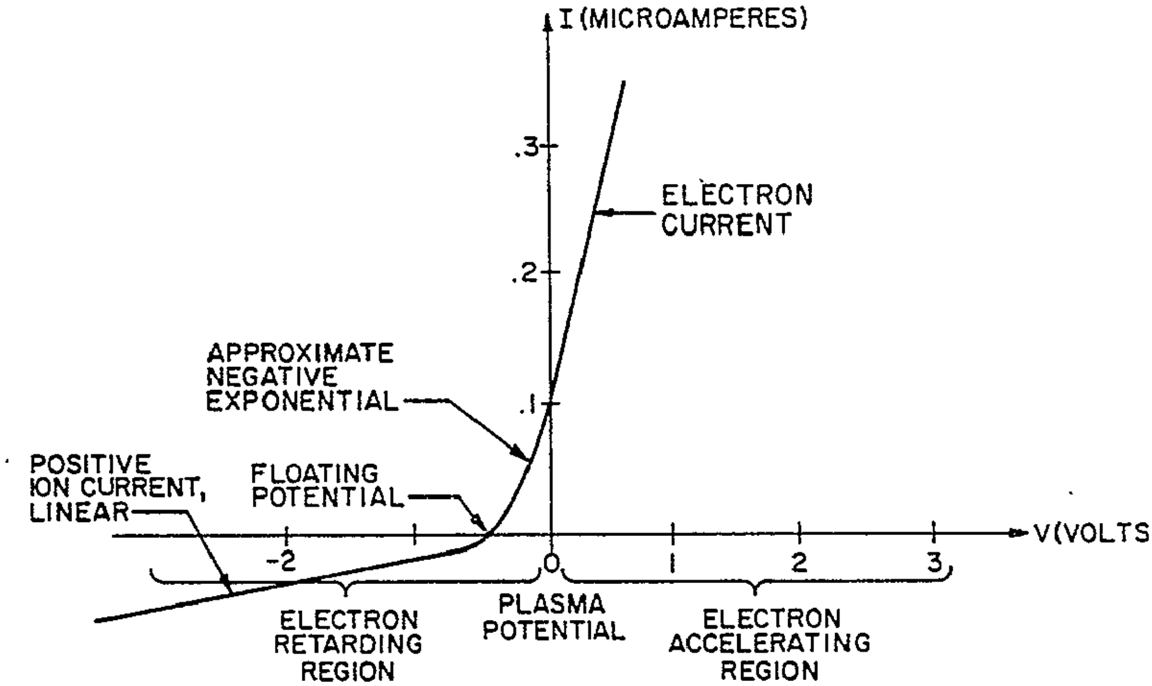


Figure 4.4 Total electrode current, including electron and positive-ion components (ZIMMERMAN AND SMITH, 1980).

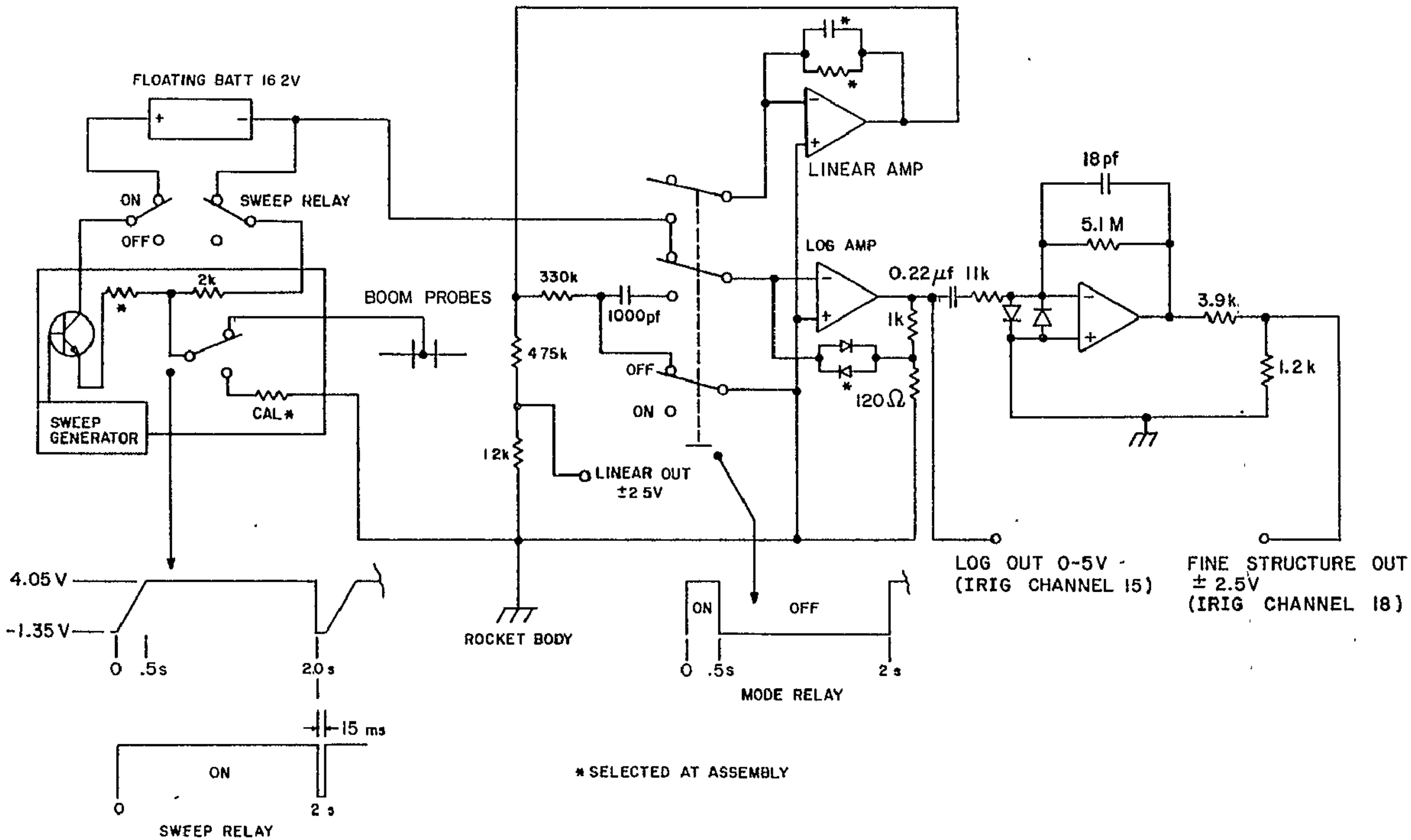


Figure 4.5 Langmuir probe instrumentation block diagram.

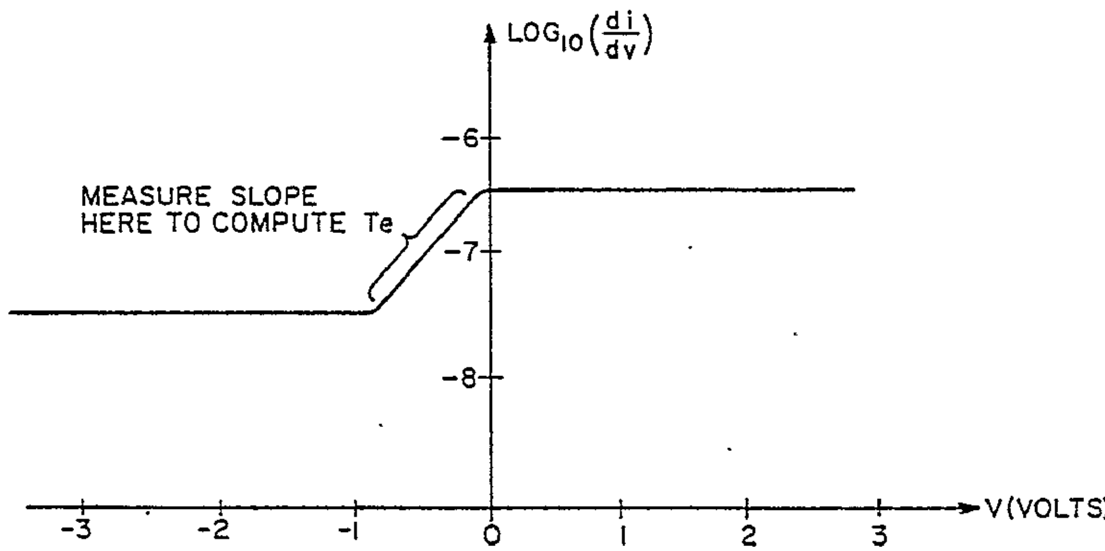


Figure 4.6 Representative graph of $\log_{10}(\frac{di}{dv})$ versus V. Temperature is determined by measuring the slope of the increasing segment of the curve. Plasma potential is assumed to be 0 Volts (ZIMMERMAN AND SMITH, 1980).

portional to the slope of the increasing part of the curve.

$$\ln\left(\frac{dj}{dV}\right) = \ln(j_o e/kT_e) + eV/kT_e \quad . \qquad (4.5)$$

Floating potential is the point at which the curve turns upward.

Electron concentration is directly proportional to the current density, j, provided V is greater than the plasma potential (Equations 4.1 and 4.4). Using 4.05 V as the fixed voltage guarantees that V will be greater than plasma potential.

The logarithmic output is capacitor coupled to a linear amplifier with a gain of 100. This separates the AC component of the log output and amplifies it by a factor of 100. This is used to determine the fine structure of the ionosphere. As the rocket passes through small-scale plasma irregularities, or if a propagating plasma wave is present in the rocket environment, its spectrum will be contained in the output of the amplifier.

Referring to Figure 4.5 again: after a period of 1.5 sec, during which the probe is held at a constant potential of 4.05 V, the double-pole relay is energized for a period of 0.5 sec by a trigger circuit which simultaneously enables the ramp generator. The probe voltage increases linearly with time over the 0.5 sec from -1.35 V to 4.05 V following a "dead time" of 15 msec. The probe current is monitored by the linear electrometer and input to the logarithmic electrometer through a differentiating capacitor. In this mode the output from the logarithmic electrometer is log(di/dV). (i = Aj, where A is the effective surface area of the probe.)

The experiment is calibrated in flight by substituting a calibration resistor for the probe electrode. This is done by a calibration relay which is energized for two periods of about 5 seconds each. This relay is controlled by two barometric switches, the first being closed for altitudes greater than 12 km, while the second is closed for altitudes less than 21 km. With these switches in series the calibration relay is energized between 12 and 21 km.

The following references may be consulted for additional information: SMITH (1969) for the electron density experiment; ZIMMERMAN AND SMITH (1980) for the electron temperature experiment; KLAUS AND SMITH (1978) for the fine structure experiment and ROTH (1982) for determination of vehicle potential.

4.3 Propagation Experiment

A radio wave passing through the ionosphere will undergo changes in amplitude and polarization due to the changing electron density of the plasma. By placing a radio receiver in a sounding rocket, the changes in the properties of the radio wave with altitude can be measured. By this method the absorption, refraction and reflection of the different modes of ionospheric propagation may be evaluated at a single frequency, and electron densities measured to a lower limit of about 10^2 cm^{-3}.

APPLETON (1932) and HARTREE (1931) developed the basic theory of electromagnetic waves propagating through a plasma contained in an external magnetic field. This applies to the earth's ionosphere above approximately 85 km (upper D region). The theory was later extended by SEN AND WYLLER (1960) to include the lower D-region.

4.3.1 Theory. A travelling wave entering the ionosphere will be split into two elliptically polarized modes, each mode with a different index of refraction. The Appleton-Hartree theory determines the modes of propagation and the indices of refraction of these modes under the following assumptions: (1) electron collisions with neutrals are independent of electron energies; (2) the medium of propagation is electrically neutral with a uniform charge distribution; (3) the magnetic field is uniform throughout the medium; and (4) the ions, because their masses are much greater than that of an electron are stationary. The theory can be further simplified by taking the wave to be travelling nearly parallel to the earth's magnetic field (the quasi-longitudinal approximation).

The refractive index, n, of the medium is given by

$$n^2 = 1 - \frac{X}{1 - j\,Z \pm Y_L} \tag{4.6}$$

where

$$X = \frac{\omega_N^2}{\omega^2} = \frac{N_e^2}{\varepsilon_o m} \cdot \frac{1}{\omega^2}$$

$$Y_L = \frac{\omega_L}{\omega} = \frac{-e\,B_L}{m\omega}$$

$$Z = \frac{\nu}{\omega}$$

and

ω ≡ angular frequency of the wave

N_e ≡ electron density

e ≡ electron charge

m ≡ electron mass

ε_o ≡ permittivity of free space

B_L ≡ component of earth's magnetic flux density along the ray path effective collision frequency of electrons with other constituents of the medium.

ν ≡ effective collision frequency of electrons with other constituents of the medium.

The $\pm Y_L$ term in Equation 4.6 yields two values for the refractive index, n, in space, giving rise to two modes of propagation. The wave polarizations are

$$R = \pm j \tag{4.7}$$

The +j polarization, indicating north into east circular polarization, is associated with the $-Y_L$ term in Equation 4.6 and is called the extraordinary mode. The -j polarization, indicating north into west circular polarization, is associated with the $+Y_L$ term in Equation 4.6 and is called the ordinary mode.

A further approximation can be made in Equation 4.6 when the local plasma frequency is much lower than the radio frequencies used in the experiment. Then

$$X \ll 1 \tag{4.8}$$

and Equation 4.6 becomes, by binomial expansion,

$$n = 1 - \frac{\frac{1}{2}X}{1 - j Z \pm Y_L} \tag{4.9}$$

Upon rationalizing, we obtain

$$n = 1 - \frac{1}{2}\frac{X(1 \pm Y_L)}{(1 \pm Y_L)^2 + Z^2} - j\frac{1}{2}\frac{XZ}{(1 \pm Y_L)^2 + Z^2} \tag{4.10}$$

where now the minus sign represents the ordinary mode and the plus sign the extraordinary mode. Equation 4.10 has the form

$$n_x = \mu_x - j\,\chi_x \tag{4.11}$$

$$n_o = \mu_o - j\,\chi_o \tag{4.12}$$

When these expressions are used in the equations for a plane electromagnetic wave (satisfying Maxwell's equation) propagating in the positive z direction, the travelling waves are

$$\vec{E}_x\,(z,t) = \vec{E}_x \exp(-\chi_x kz) \exp[j(\omega t - \mu_x kz)] \tag{4.13}$$

$$\vec{E}_o\,(z,t) = \vec{E}_o \exp(-\chi_o kz) \exp[j(\omega t - \mu_o kz)] \qquad (4.14)$$

where k is the free space propagation constant.

Since any wave can be described as the sum of two circularly polarized waves, (one right-handed and one left-handed), the plane of polarization of the wave will rotate due to the differing refractive indices. This is Faraday rotation.

For a rocket moving in the +z direction,the Faraday rotation rate, F, will be given by

$$F = \frac{360}{2\pi}\ \frac{k}{2}(\mu_o - \mu_x)v \qquad (\text{deg-s}^{-1}) \qquad (4.15)$$

where v is the rocket velocity in $m\text{-}s^{-1}$. Since

$$k = \frac{\omega}{c} \qquad (4.16)$$

Equation 4.15 becomes

$$F = \frac{180\ f}{c}\ (\mu_o - \mu_x)v \qquad (\text{deg-s}^{-1}) \qquad (4.17)$$

It follows from Equation 4.6, 4.10 and 4.17 that Faraday rotation rate is directly proportional to electron density.

The differential absorption rate, A, for a rocket moving in the +z direction is given by

$$A = 20\ \log_{10}\{\exp[-k(\chi_o - \chi_x)v]\} \qquad (\text{dB-s}^{-1}) \qquad (4.18)$$

which reduces to

$$A = 8.686\ \frac{2\pi f}{c}\ (\chi_x - \chi_o)v \qquad (\text{dB-s}^{-1}) \qquad (4.19)$$

It follows from Equations 4.6, 4.10 and 4.19 that the differential absorption rate is directly proportional to electron density.

The approximations used in the preceding analysis are not always valid, and, therefore, the full theory is used in the actual data analysis. Additionally, the Appleton-Hartree theory is not acceptable for the lower D region where collision frequencies are large. Therefore, the generalized Sen-Wyller equations are used in the calculations (SLEKYS AND MECHTLY, 1970; GINTHER AND SMITH, 1975).

4.3.2 Implementation. Because the propagation experiment was designed in the middle sixties, greater real-time processing is included in order to simplify post-flight processing. The experiment measures both Faraday rotation and differential absorption to obtain electron density information. To accomplish this the experiment uses a transmitted radio signal which easily allows both effects to be identified. This is done by independently generating two signals having the polarizations of the ionospheric ordinary and extraordinary modes and then combining them to form a composite transmitted signal. Figure 4.7 graphically describes the signals which are transmitted from a location near the launch site. The ordinary mode rotates north into west at a frequency of $f_o = f_c - 250$ Hz, where f_c is the center frequency. The extraordinary mode rotates north into east at a frequency of $f_x = f_c + 250$ Hz. The two modes differing in frequency by 500 Hz, and in power by 10 dB, form a resultant ellipse (Figure 4.7) whose axis rotates in free space at 250 revolutions per second, north into east.

As this resultant signal propagates through the ionosphere, Faraday rotation will cause the phase of the polarization ellipse to increase with altitude according to the electron density profile. In the reference plane of the moving rocket this change in phase appears as a change in frequency of the polarization rotation rate. By measuring the frequency of polarization rotation and extracting the contribution due to Faraday effect

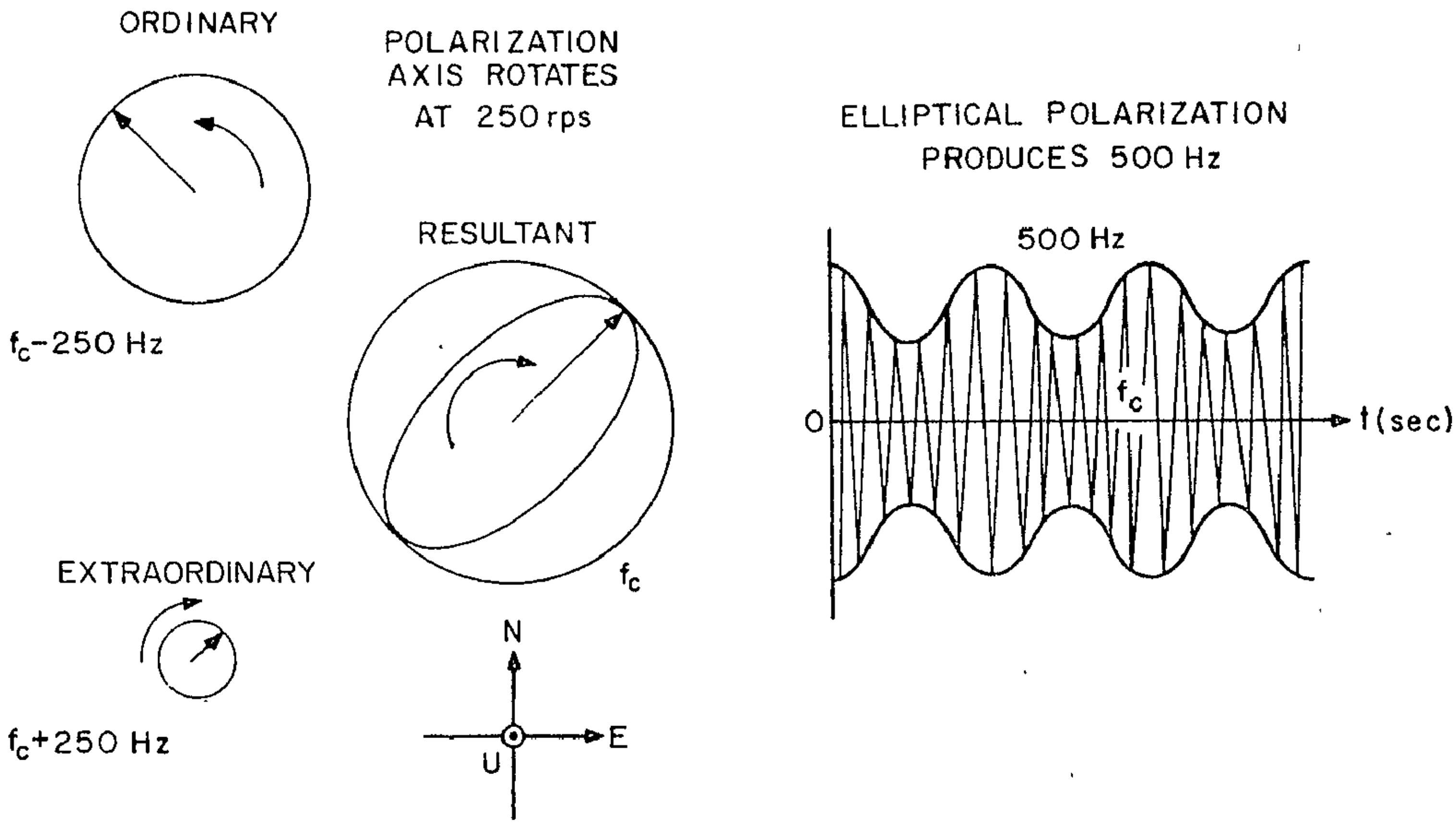


Figure 4.7 Generation of polarization ellipse (SLEKYS AND MECHTLY, 1970).

the electron density can be determined.

The extraordinary mode of the resultant signal is absorbed about six times more rapidly than that for the ordinary mode. Differential absorption is represented by the increase of extraordinary transmitted power required to maintain some fixed amplitude modulation ratio between the two modes, as seen at the output of the rocket receiver. The 10 dB power ratio used in the experiment produces a 31.6% modulation of the signal received at the rocket. Hence increasing the extraordinary mode power level (to maintain the 10 dB ratio) as the rocket ascends gives a measure of the differential absorption in dB per second from which the electron density can be determined.

Figure 4.8 is a block diagram of the radio propagation experiment. The two exciters, designated X and O, are crystal controlled oscillators operating at f_c + 250 Hz and f_c - 250 Hz, respectively. The exciters feed a phase detector the output of which is a 500 Hz difference signal. This difference signal is sent to the telemetry station where it is recorded to be used later to determine Faraday rotation rate. The attenuators, also fed by the oscillators, are controlled by the feedback signal from the rocket; this output is then divided between a variable phase shifter and a variable attenuator for the generation of circular polarization. These signals are then added, amplified and fed to the antenna array, which consists of four horizontal half-wave dipoles elevated one quarter wavelength above the ground. The two magnetoionic modes of polarization are thus generated and transmitted to the rocket.

The rocket payload equipment required to receive the composite transmitted signal includes a linearly polarized ferrite loop antenna feeding a receiver whose detector is of the envelope detection type. The receiver

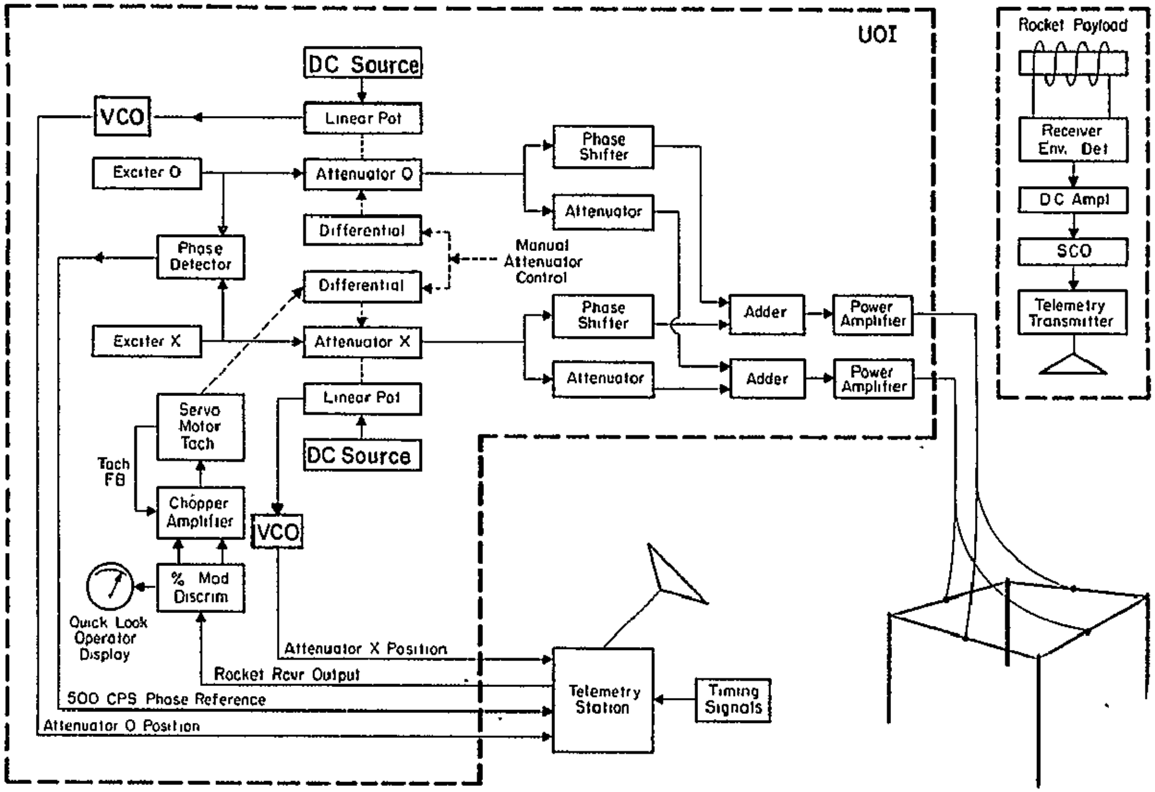


Figure 4.8 Radio propagation experiment system block diagram.

output consists of DC and 500 Hz signals having an amplitude ratio corresponding to the 10 dB power ratio of the two modes. The output of the receiver is telemetered to the ground where it is recorded and also sent to the transmitter van.

At the van, the percent modulation of the rocket receiver output is measured. Under conditions of differential absorption the percent modulation will fall below its nominal 31.6% value. When this is observed at the van, the extraordinary power is automatically increased via the servo motor on the X attenuator to maintain the 31.6% modulation value. Power levels are transmitted via a frequency modulated data link to the telemetry station where they are recorded. A potentiometer attached to the attenuator varies the DC voltage applied to a voltage controlled oscillator thus assigning to each attenuator position a selected frequency (Figure 4.9).

The complete propagation experiment consists of two systems operating simultaneously, but on different radio frequencies. The operating frequencies are chosen to provide significant daytime differential absorption while still leaving sufficient signal for Faraday rotation measurements.

The 2 to 5 MHz range has been found to adequately satisfy the above requirements. Typically one system operates at 2.225 MHz, while the other operates at either 3.385 or 5.040 MHz. The criterion for choosing between 3.385 or 5.040 MHz is based upon which frequency combination is expected to give the best compromise between measurement sensitivity and highest attainable measurement altitude. Measurement sensitivity is inversely related to the square of the frequency; however, lower frequencies have lower reflection heights and thus lower maximum measurement altitudes. In general 3.385 MHz is used at night and 5.040 MHz during the day.

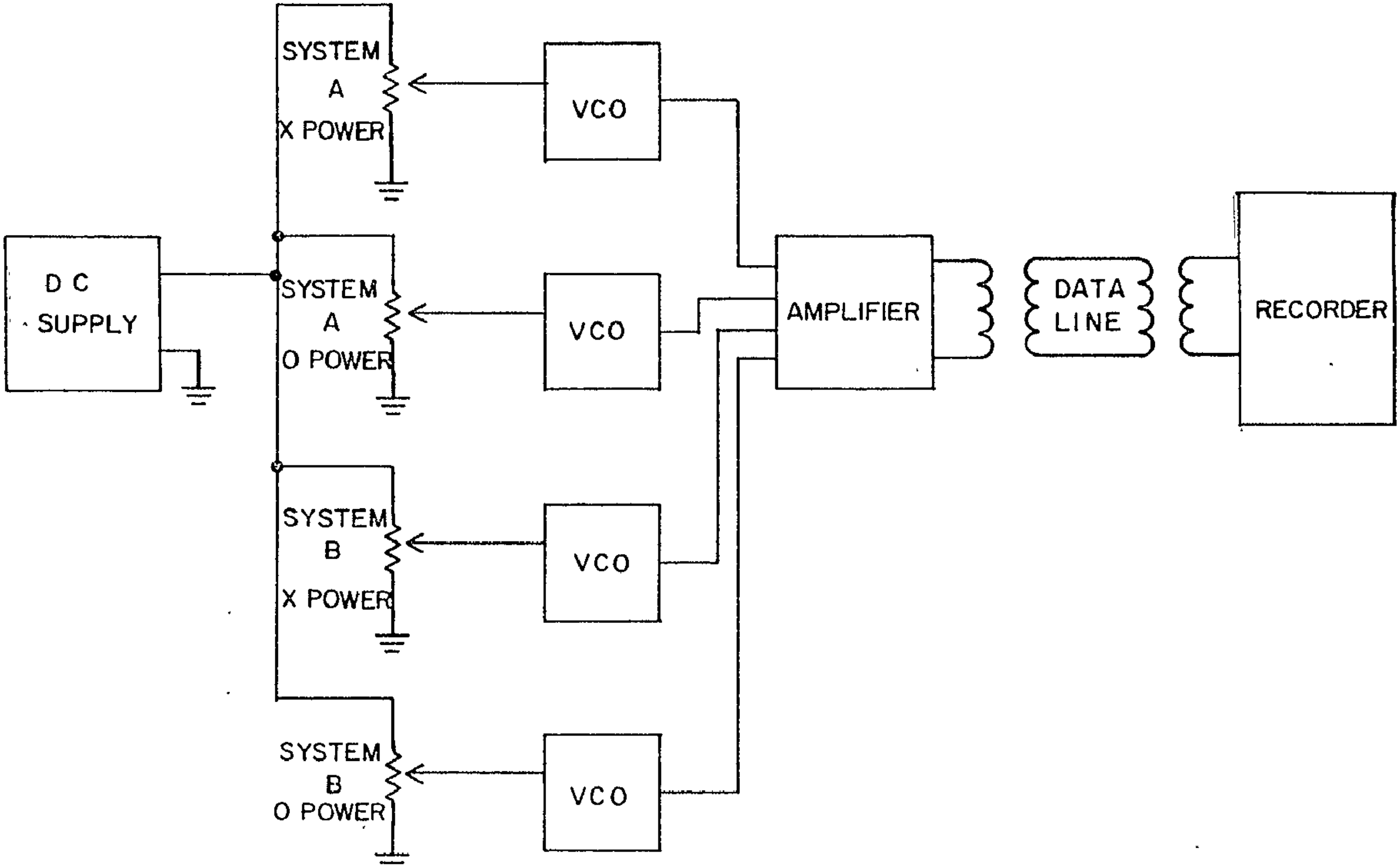


Figure 4.9 Block diagram of frequency modulated data link (GINTHER AND SMITH, 1975).

For more information on the radio propagation experiment, the following University of Illinois Aeronomy Laboratory publications may be consulted: GINTHER AND SMITH (1975) and FILLINGER ET AL. (1976) contain a more detailed description of the Appleton-Hartree and Sen-Wyller magnetoionic theories; the rocket receiver schematic is shown in EDWARDS (1980); and a detailed analysis of the receiver output is given in GILCHRIST AND SMITH (1979). A general outline of the original experiment (as it existed in 1970) is described in SLEKYS AND MECHTLY (1970).

4.4 Energetic Particle Experiment

Energetic charged particles propagating within a solid lose kinetic energy through lattice interactions. Lattice electrons are lifted from the filled valence bands into the essentially empty conduction band. By applying a potential gradient within the material an electric current is established for a short period. The total charge carried by this current is proportional to the energy of the particle. After conversion to a voltage the pulse amplitude is recorded.

Different types of energetic charged particles (e.g., electrons and protons) will differ in their rate of energy loss while travelling through a solid. Identification of particles can be accomplished by varying the thickness of the solid through which the particles travel and then comparing energy losses.

4.4.1 Theory. There are many methods of particle detection. Figure 4.10 illustrates the energy range capabilities for various detectors. The channeltron (channel multiplier) is the best for detection of low energy particles. However, there are complications affecting its use in rockets: (1) the gain is pressure sensitive due to the ionization of the neutral residual gas; (2) the presence of high voltage required for operation;

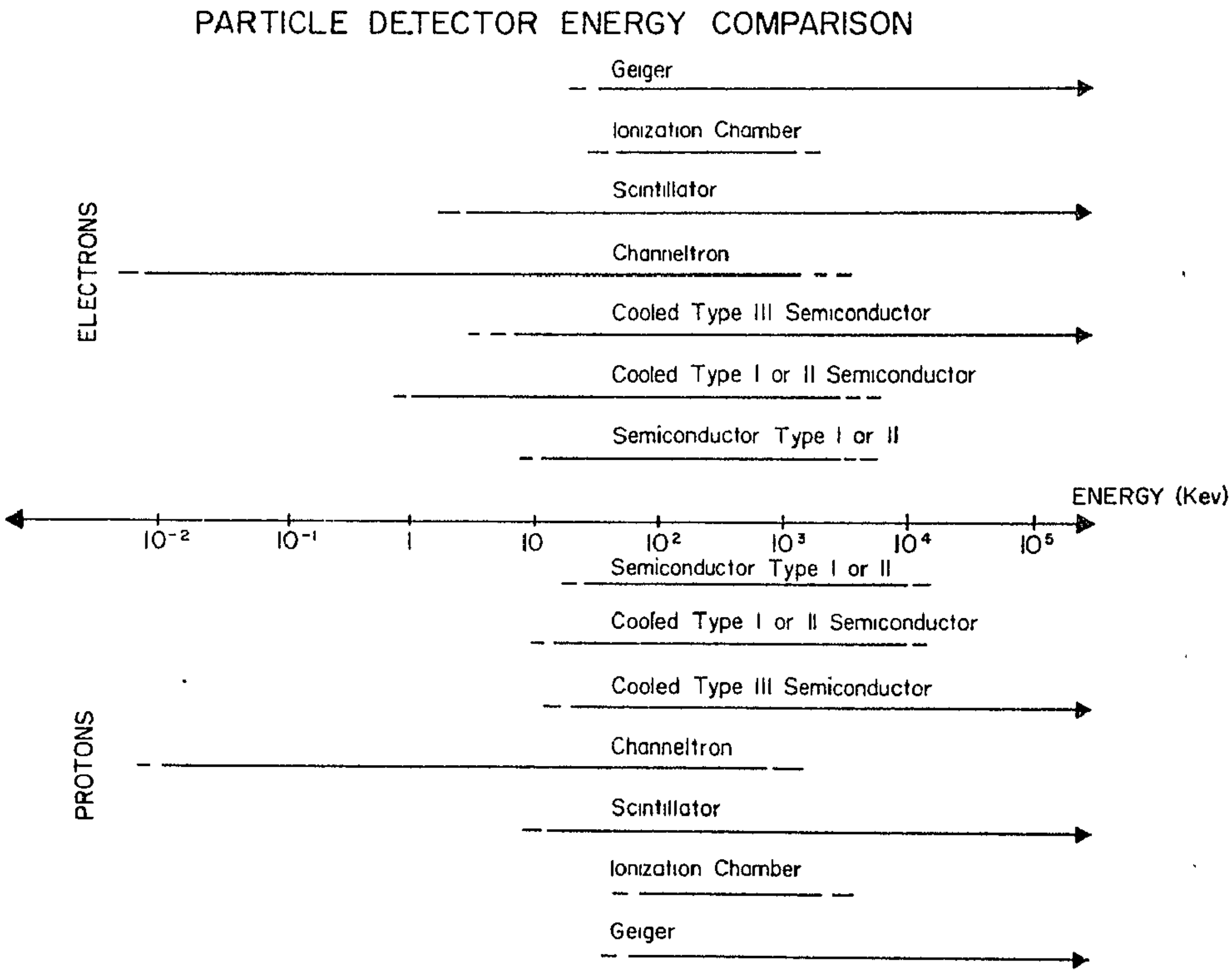


Figure 4.10 Sensitive energy range for various rocket-borne particle detection schemes. The energy comparison is made for electrons in the top half and for protons in the lower half.

(3) the possibility of UV contamination; and (4) the small geometrical factors.

Gaseous detectors may be classified into three modes of operation: ionization, proportional, and Geiger. These detectors are well suited to rocket-borne experiments. However, the main disadvantages are: (1) sensitivity with respect to where the ionization track takes place with the gas; (2) gas purity; (3) associated deadtime due to slow ion mobilities; and (4) high voltage requirements.

The particle detector used for this experiment is based on a solid-state surface barrier detection system. The advantages of this detector type for rocket applications are:

(1) better statistics and resolution result since many more charge carriers are released for a particular incident energetic particle, (e.g., an ionization event requires 35 eV, typically, for a gas and 3.5 eV for silicon).

(2) insensitivity to UV or low energy X rays since operated in a non-avalanche mode.

(3) very short deadtimes: of the order of a few nanoseconds. The collected charge is independent of the location of the ionization event.

(4) the detectors are rugged, compact, easily mounted and use low voltages.

The primary disadvantages are: the relatively weak signals requiring extremely sensitive electronics; and a 10 keV low energy threshold.

The surface-barrier diode, sometimes called a Schottky barrier diode, consists of a metal layer deposited onto an n-type or p-type silicon crystal. A diode is formed by using gold with p-type silicon or aluminum with n-type silicon. A depletion layer forms in the silicon at the barrier, as shown in Figure 4.11. The detector is used with a reverse bias on the

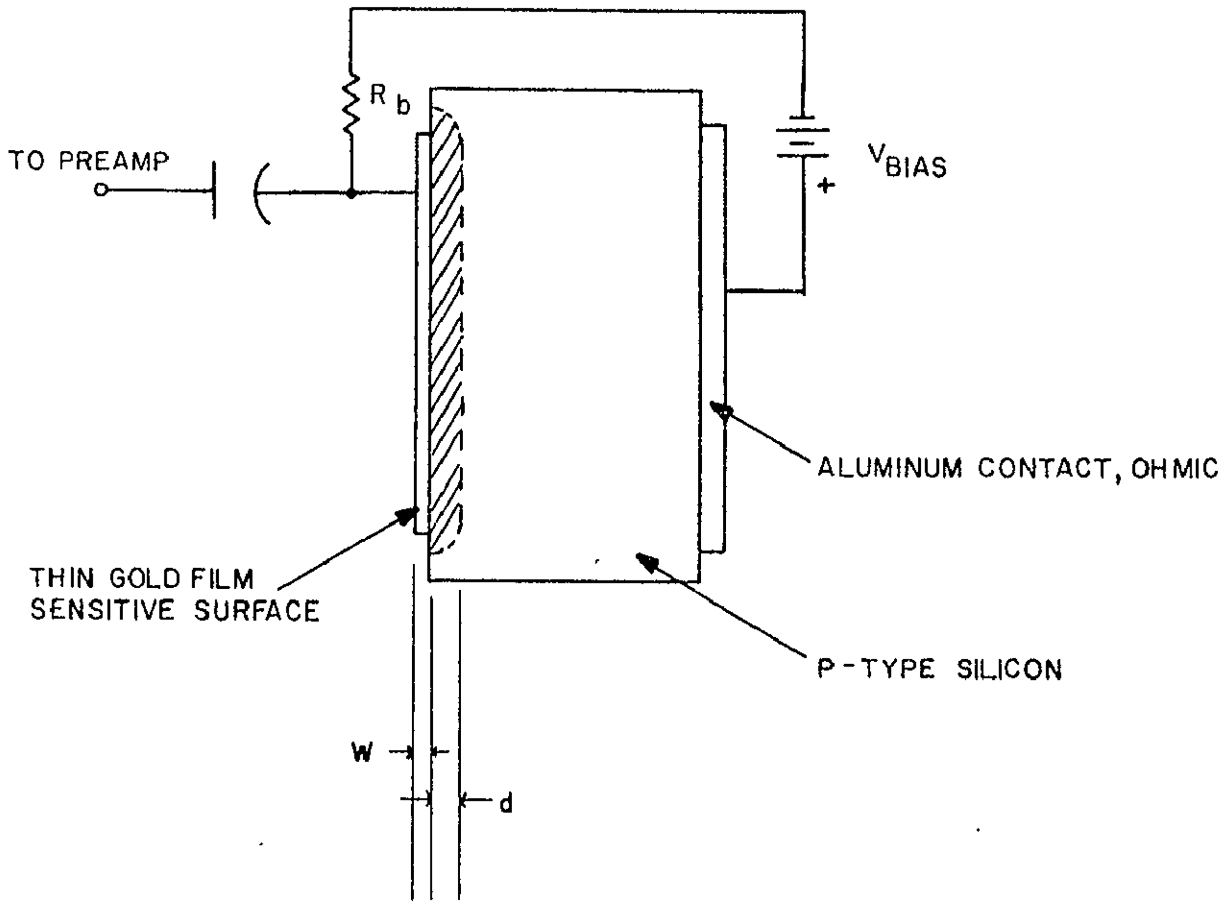


Figure 4.11 Illustration of surface barrier diode. The width of the surface metal is indicated by W. The depletion layer width (sensitive depth) is indicated by d and is on the order of 100 μm.

junction. This ensures rapid removal of the electron-hole pairs from the depletion layer. The bias (normally 100 V) is not large enough to cause avalanche breakdown.

The width of the depletion region with the reverse bias applied is an important design consideration. The incident particle must lose all of its energy within the depletion region in order to yield valid energy information. The range of energetic particles in silicon and germanium for given energies is well documented. With this knowledge one can determine the maximum measurable energy for a detector with a specified maximum bias. The width of the depletion layer is also important in determining the junction capacitance, which is approximated as a parallel-plate capacitor. The maximum field strength allowed in silicon (to avoid avalanche breakdown) is related to the depletion layer width and the applied bias. The depletion layer thickness for a reverse bias voltage V is given by

$$d = [2\ \varepsilon_r\ \varepsilon_o\ \mu\ \rho(V + V_o)]^{\frac{1}{2}} \tag{4.20}$$

where

ρ is the resistivity of the doped semiconductor

μ is the electron mobility of the doped semiconductor

V_o is the contact potential

ε_r is the relative dielectric constant of the material

ε_o is the permittivity of free space.

Using the formula for a parallel-plate capacitor the junction capacitance for a detector with surface area A is

$$C_J = A\ \varepsilon_r\ \varepsilon_o[2\ \varepsilon_r\ \varepsilon_o\ \mu\ \rho(V + V_o)]^{-\frac{1}{2}} \tag{4.21}$$

C-2

There is a layer on the surface of a semiconductor detector where energy lost by the incident particle will not contribute to the measured energy. This is the dead zone. For a surface-barrier diode the dead-zone thickness is approximately equal to the thickness of the metal layer (see Figure 4.11). The amount of energy lost in the dead zone depends upon the distance a particle must travel before it reaches the depletion region. Particles with a large angle of incidence (relative to the normal) travel a greater distance through the dead zone and lose more energy than particles entering normal to the detector. The dead zone introduces a limit on the energy resolution and some uncertainty in detected energy because of variation in angle of incidence.

The dead zone can be used to aid in determining the type of charged particle present if more than one detector is used. The energy lost per unit distance travelled varies with the particle mass, charge and energy according to the formula:

$$\frac{dE}{dx} \propto \frac{Z^2 M}{E} [\ln \frac{E}{M} + \text{constant}] \quad , \qquad (4.22)$$

where E is the kinetic energy of the charged particle; x is the distance measured along the particle track; Z is the charge of the particle; and M is the mass of the particle. If a detector is used with a very thin dead zone, then very little energy is lost in the dead zone for any particle. If a detector with a thick dead zone is used, then the output would be similar to that of the detector with the thin dead zone if electrons are present, but much smaller if heavy particles such as protons or oxygen ions are involved. By including two detectors, one with a thin metal layer (40 $\mu g\ cm^{-2}$ Al) and one with a thick metal layer (100 $\mu g\ cm^{-2}$ Al), it is possible to distinguish electrons from heavier particles.

The output pulse of a detector is determined by the charge collection time and series impedance (resistive and capacitive) of the detector circuit during the pulse rise time and the time constant of the input circuit of the associated electronics during the pulse fall time. The charge collection time is theoretically limited by the deceleration time of the incident particle. However, the depletion layer thickness, electric field strength, and carrier mobility are also important. A high bias voltage and a low resistivity material (to maximize the electric field strength) are necessary for a fast rise time. For a short time the electron-hole pairs balance the effect of the applied electric field; the duration of this effect is called the plasma time or plasma effect. The net result is a lengthening of the rise time.

The amplitude of the charge pulse will be reduced if some of the electron-hole pairs recombine as they are being swept out of the depletion region. The probability of recombination can be reduced by increasing the bias voltage, thus reducing the collection time. More serious than electron-hole recombination is trapping of charge carriers due to lattice defects or impurities. These tend to be non-uniform and the resulting charge pulse may be dependent on the path taken by the particle. The effect of trapping can be a multiple-peaked signal resulting from a monoenergetic stream of particles.

Lattice defects caused by high energy particles are referred to as radiation damage. As a detector is used the lattice defects caused by radiation damage become more pronounced. The resolution of the detector decreases and becomes voltage dependent. With the short exposure time and relatively low energy particles encountered in rocket-borne applications the radiation damage is small. Surface-barrier detectors generally have lower levels of impurities than diffused detectors. This results in a higher resolution for barrier detectors due to fewer trapping centers.

4.4.2 Implementation. A complete block diagram of the energetic particle experiment is represented in Figure 4.12. The current introduced into the preamplifier by the detector from an incident energetic particle is integrated by applying capacitive feedback (i.e., charge amplified). The resulting charge pulse is routed through a detector selector switch, which switches at apogee. The pulse is shaped (to minimize noise) and further amplified before being height discriminated and counted. Finally, the digital counter value is converted to analog form for telemetering to the ground.

Figure 4.13 is a cut-away view of the collimator/detector mount. The collimator is used to define the angular response of the detector. By reducing the detector's view angle, it is possible to measure the direction of arrival of the incident particles, and problems caused by differing travel distances in the detector dead zone are reduced.

The geometrical factor of the detector (and collimator) is defined by

$$G \equiv N/\Phi \qquad (4.23)$$

where G is the geometrical factor; N is the detector count rate per second; and Φ is the number of particles per square centimeter per second per steradian. It has been assumed that the incident flux is isotropic over the solid angle for which the detector is sensitive. The geometrical factor is a function of t, r, and R and is related to the effective area, $A(\theta)$, where A is a function of the angle of incidence, θ.

Usually the geometrical factor needs to be found numerically because of the complicated dependence of the effective area, A, on θ. The geometrical factor of the eclipse payload detectors is 0.05 cm^2 ster.

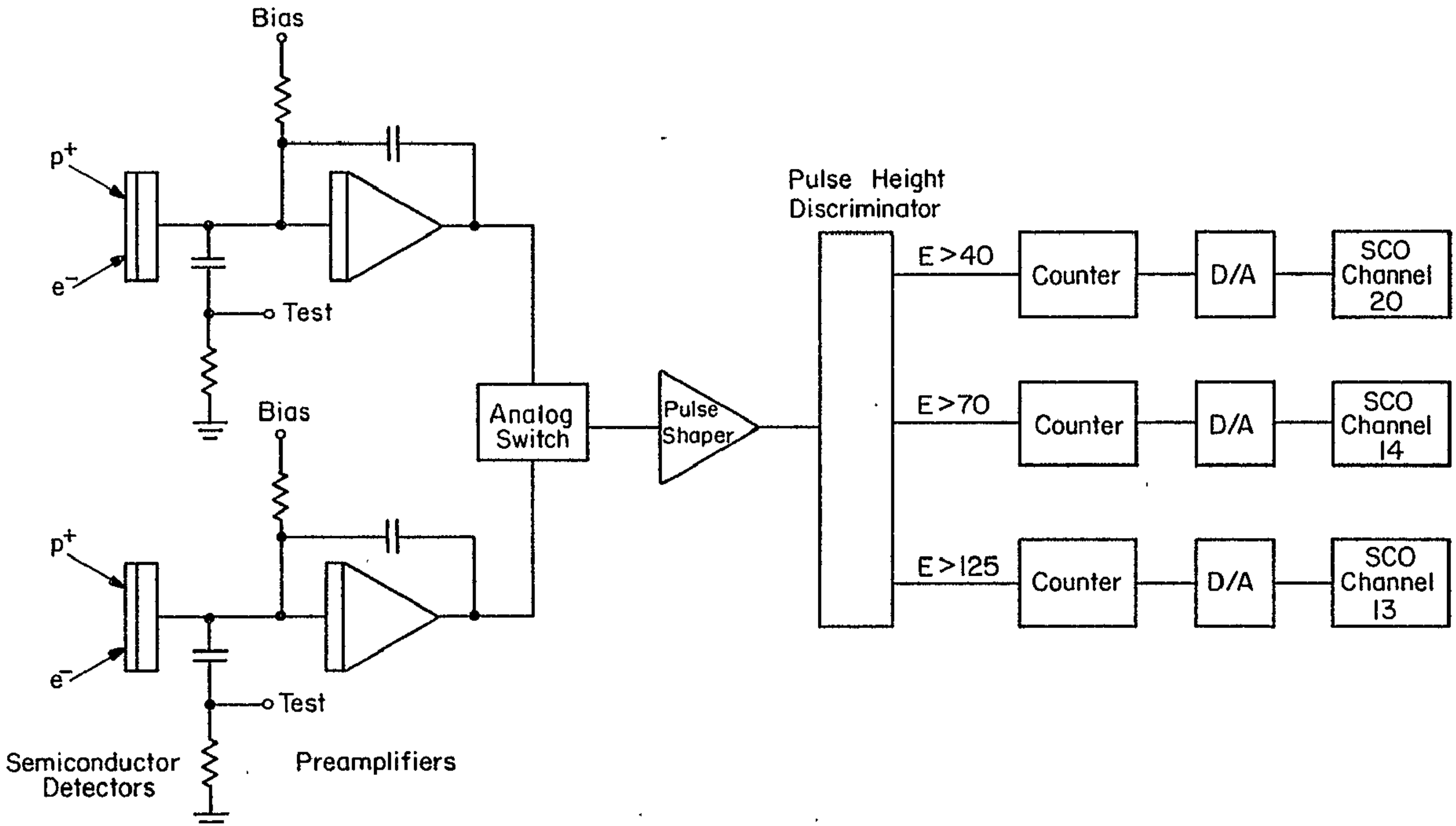


Figure 4.12 System representation of the solid-state energetic particle detector experiment.

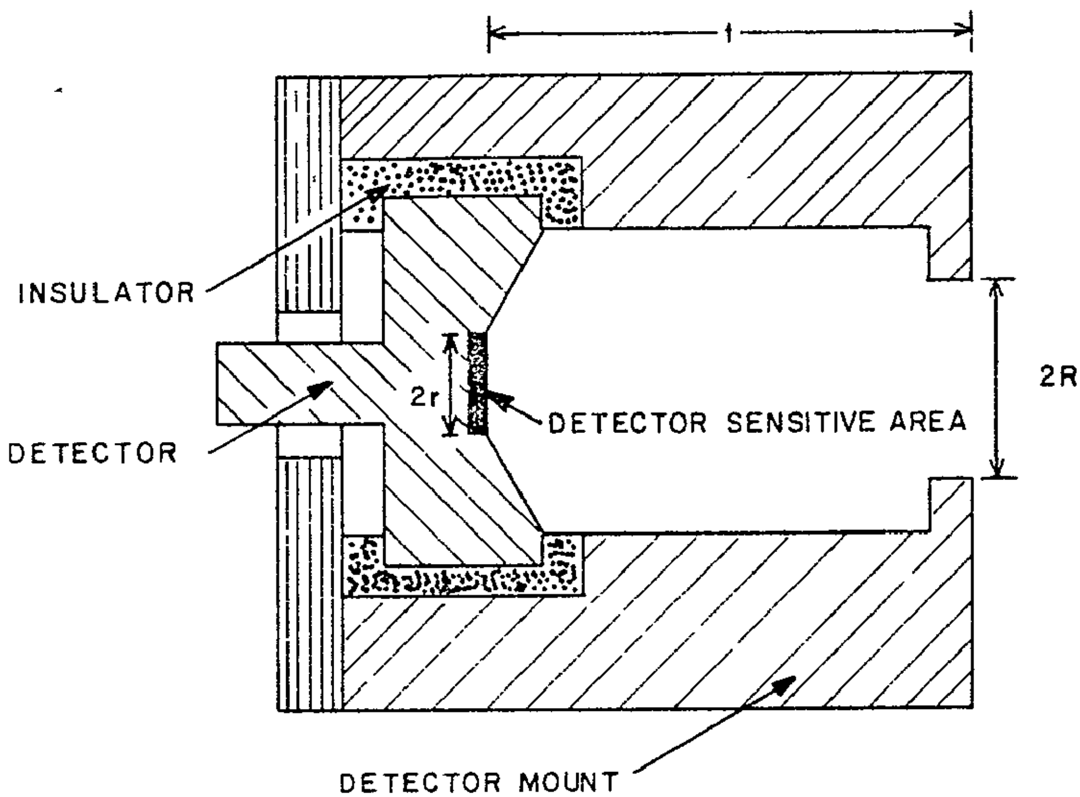


Figure 4.13 Cutaway view of collimator/detector mount.

The angular response of the detector and collimator assembly can also be found experimentally by plotting the count rate as a function of the angle of incidence for a monoenergetic beam of constant particle flux. One definition of angular response is the angle at which the count rate is half of the maximum rate (full width at half maximum).

The preamplifier is a high-gain low-noise charge-amplifier with a junction field effect transistor (JFET) used in the first stage. High gain is required to confine the noise injected into the system to the detector and the first stage. Similarly, low noise is necessary to minimize the noise that does enter the system and degrade performance.

A charge-amplifier configuration is used to make the preamplifier less sensitive to variations in detector capacitance. The detector is primarily capacitive; the value of the capacitance varying with the temperature of the detector and with the applied bias. Additionally, the preamplifier has a high input impedance so that the charge from the detector appears across the input capacitance and is not dissipated in the shunt resistance across the detector.

Because the detectors are mounted on the booms a longer cable than that normally used is needed between the detector and the preamplifier, presenting a higher-than-normal cable capacitance. This contributes to a slightly poorer noise performance and a 20 keV detection limit.

The two charge-preamplifiers are connected to an analog switch at the input to the pulse shaping network. The analog switch selects either detector 1 or detector 2 as the input to the pulse shaper. Detector 1 has a 40 $\mu g\ cm^{-2}$ Al surface and is connected to the pulse shaper during ascent. Detector 2, which has a 100 $\mu g\ cm^{-2}$ Al surface, is connected to the pulse shaper during descent.

The pulse shaper consists of four buffer amplifiers connected in series through resistor-capacitor (RC) networks. The first RC network performs high pass filtering of the signal between buffer amplifiers 1 and 2. The remaining two RC networks (between buffer amps 2 and 3 and buffer amps 3 and 4) perform low-pass filtering of the signal. Together these RC networks form a band-pass filter. The filter serves to remove the dc component of the signal and to adjust the length of the output pulse so that two pulses occurring nearly simultaneously will not overlap. The filter also limits the rise time of the pulse and the sharpness of the pulse peak to insure proper operation of the pulse-height discriminator.

The pulse-height discriminator consists of three series connected voltage comparators and their associated monostable multivibrator. Each comparator is referenced at a different voltage from a resistive voltage divider network. Hysteresis is included to reduce the sensitivity of the transition region to approximately 3 mV. The monostable multivibrator is used for triggering and predetermining the deadtimes specified for each channel. When a negative-going signal is received from the comparator the monostable triggers and outputs a pulse of fixed length. The output is unaffected by additional input transitions during the period of the monostable output pulse. This eliminates effects from multiple triggering and defines the pulse shape which drives the succeeding counter stage. The pulse length chosen is determined from the data capability (bandwidth) of the telemetry channel being used for this energy range.

The shaped pulses are counted in a 4-bit binary counter and paralleled to a digital-to-analog converter (DAC). The output from the DAC is interfaced with the telemetry voltage controlled oscillators (VCO's) by an operational amplifier. The operational amplifier functions as a current to voltage converter.

The energetic particle experiment is described in much greater detail in VOSS AND SMITH (1974) and in FRIES ET AL. (1979).

5. DATA PROCESSING

5.1 Introduction

Data processing techniques and computer programs will be discussed in this chapter. Section 5.2 discusses the creation of the digital data tapes used in processing the flight data. The remaining sections give details of the analysis procedure for each experiment. Each subsection describes a computer program used in the analysis. There are two programs for processing the probe data; three programs for the Faraday rotation data; four programs for the differential absorption data; three programs for producing a final electron density profile; and one program for processing the energetic particle data.

The analysis is sequential within each section. Any manual analysis is described within a computer-program subsection, the program subsection for which the analysis generates input.

Only the programs which will continue to be used after processing the eclipse data are described in detail, with a flowchart to aid in following program steps. Although not all the programs are described in detail, all are listed with documentation in the appendices.

Included in each program appendix are the operating system commands used to run the program. The only statement necessary to run the programs which has been left out is the password statement. The IBM jobs require a password on statement three of the Job Control Language (JCL) listing, following CODE= . The Cyber jobs require a password statement to be inserted after the SIGNON(3MIKEKM) statement, except, of course, for the two time-sharing programs.

Data from flight 18.1020 are used as an example of data processing throughout this chapter and the appendices.

Appendices XIV and XV contain some useful IBM and Cyber procedures. Appendix XIV lists two sytem procedures (one IBM, one Cyber) for examining the contents of a tape. The IBM TPSNIF procedure is useful for examining NASA data tapes. It lists the length of each record on the tape in IBM words.

The Cyber EXAMINE procedure works best with Cyber created tapes, multiple file tapes, and non-binary data tapes. It lists the number of records per file, maximum record length per file, type of data in file, together with many other diagnostics.

Three system procedures (two IBM, one Cyber) for copying one tape to another are listed in Appendix XV. Occasionally a tape which generates many read parity errors is encountered, especially older NASA data tapes when used on the Cyber. Many times these parity errors can be eliminated by creating a new tape. Appendix XV shows various methods of duplicating tapes.

Many of the analysis procedures in this chapter require conversion of the IBM format data tape to Cyber format. A University of Illinois Computing Service Office (CSO) routine is used to perform this conversion on the Cyber. Appendix XVI contains the CSO documentation for using Fortran callable subroutine GBYTES, which performs the conversion.

5.2 Digitization

The undiscriminated analog recording of the flight data (see section 4.1) is used to create a digital version of the flight data. When the analog tape is played back, selected channels are discriminated, digitized and recorded in digital format on a magnetic tape. It is this digital tape which is used by the Aeronomy Laboratory to analyze the flight data.

The digitizer operates at a nominal clock rate of 5 kHz and is preceded by a five-channel commutator. A selected data channel (see Table 4.1) will be low pass filtered, sampled, digitized (with an A/D converter resolution of 12 bits) and recorded every fifth clock pulse. This results in a digitization rate of 1 kHz per data channel. The digitizer clock rate can be increased to 25 kHz to provide an even greater sampling rate.

Table 5.1 lists the channel assignments and digitizing rates for three of the post-flight tapes. The cut-off frequency of the low pass filter used with the IRIG channel to be discriminated is given in parentheses. MPX stands for multiplex and refers to an additional set of IRIG channels multiplexed onto the analog tape. Note that each of these multiplexed channels corresponds to one of the UI van signals transmitted to the NASA telemetry station for recording (see section 4.3.2, Figure 4.8). One of these multiplexed channels also contains the time information (universal time derived from WWV).

The digital tapes are recorded at 800 bpi in a nine-track format. Each word contains 16 bits, the equivalent of an IBM half-length integer. Alphanumeric information is encoded in EBCIDIC. Figure 5.1 shows the organization of the tape. Each tape consists of one file with records of various lengths.

The first four records are 45 words long. These are the header records and contain tape identification information. Following the header records are five calibration records, each 1005 words in length. These records contain the digital values corresponding to the five analog telemetry levels: 0 V, 1.25 V, 2.50 V, 3.75 V and 5 V. Data are recorded in the remaining records.

A data record is 2008 words long and is structured as in Figure 5.2. Each of these records contains five channels of data running cyclically from

Table 5.1 Nike Tomahawks 18.1020, 18.1021 and 18.1022 post-flight tape requirements.

Post-Flight Tape #1: Faraday Rotation

This tape shall contain data from the following five channels, sampled sequentially:

1.	Receiver No. 1 Output	(IRIG Ch. 17, 600 LP filter)
2.	System No. 1 500 Hz Reference	(MPX #2, IRIG Ch. 17, 600 Hz LP filter)
3.	Receiver No. 2 Output	(IRIG Ch. 16, 600 Hz LP filter)
4.	System No. 2 500 Hz Reference	(MPX #2, IRIG Ch. 16, 600 Hz LP filter)
5.	Magnetic Aspect Sensor	(IRIG Ch. 9, 25 Hz filter)

Sampling rate: 25 kHz (5000 samples/channel/second), synchronized to the 100 kHz reference.

Data recording shall commence not later than T-30 sec and shall continue until 300 seconds after launch. If necessary, the data may be recorded on consecutive digital tapes.

Post-Flight Tape #2: Probe and Differential Absorption

This tape shall contain data from the following five channels, sampled sequentially:

1.	DC Probe, Log	(IRIG Ch. 15, 450 Hz LP filter)
2.	System No. 1 Extraordinary Power	(MPX #3, IRIG Ch. 6, 6 Hz LP filter)
3.	System No. 2 Extraordinary Power	(MPX #3, IRIG Ch. 4, 6 Hz LP filter)
4.	System No. 1 Ordinary Power	(MPX #3, IRIG Ch. 7, 6 Hz LP filter)
5.	System No. 2 Ordinary Power	(MPX #3, IRIG Ch. 5, 6 Hz LP filter)

Sampling rate: 5 kHz (1000 samples/channel/second), synchronized to the 100 kHz reference.

Data recording for post-flight tape 2 shall commence prior to the pre-flight University of Illinois power calibrations (which start at 1650 UT) and shall continue until the post-flight calibrations (which start at 1700 UT) are completed. In the event of a long delay between the pre-flight calibrations and the time of launch, digital recording may be interrupted after the calibrations and resume 10 seconds prior to launch.

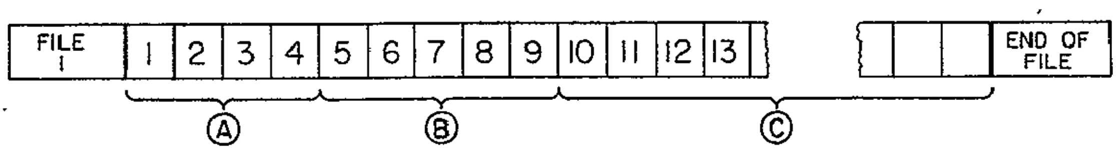
Post-Flight Tape #4: Particle Detectors

This tape shall contain data from the following five channels, sampled sequentially:

1.	Geiger counter	(IRIG Ch. 21, 2500 Hz filter)
2.	EPS output #1	(IRIG Ch. 20, 5000 Hz filter)
3.	EPS output #2	(IRIG Ch. 14, 1650 Hz filter)
4.	EPS output #3	(IRIG Ch. 13, 1088 Hz filter)
5.	Magnetic aspect sensor	(IRIG Ch. 9, 25 Hz filter)

Sampling rate: 25 kHz (5000 samples/channel/second), synchronized to the 100 kHz reference.

Data recording shall commence not less than 10 seconds prior to launch and shall continue until T+300 sec.



Ⓐ {Four header records of length 45 words containing tape identification, e.g., AMQ1 NIKE APACHE 14.532.

Ⓑ {Five calibration records of length 1005 words. These records are the digital levels corresponding to the five analog telemetry levels LBE(-7.50%), LBH(-3.75%), BAND CENTER, UBH(+3.75%), and UBE(+7.50%).

Ⓒ {Data records of length 2008 words.

Figure 5.1 Organization of data on a digital tape.

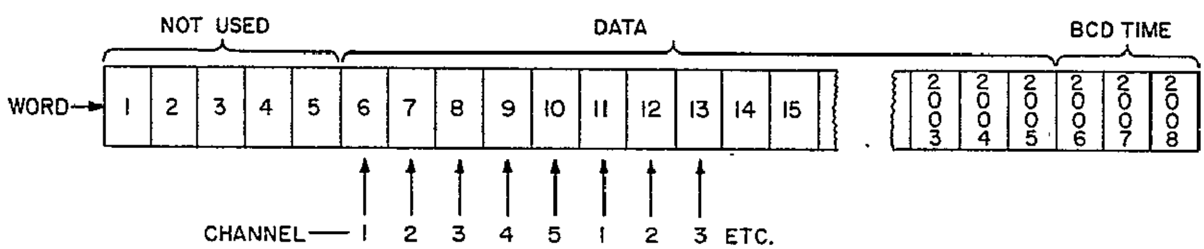


Figure 5.2 Organization of information within one data record.

word 6 to word 2005. Words 1 through 5 are not used. Words 2006, 2007 and 2008 contain binary-coded decimal time as explained in Table 5.2. The time recorded in these three words is the time of word 6 in the following record. Note that there is a constant increment of time between any two adjacent data words, even between word 2005 of one record and word 6 of the following record.

The time which is recorded in the last three words of a data record is derived from the time recording placed on the analog tape. The analog time recording consists of binary encoded universal time using a 1000 Hz carrier. When decoding, the 1000 Hz carrier provides a time signal accurate to 0.001 second. Universal time is the reference for the analog recording.

The channel on the analog tape containing the time information is fed into a time decoder/encoder which decodes the analog recording and encodes for digital recording. When the digitizer begins sampling the first data point of a record, a pulse is sent to the time decoder/encoder requesting the time. This time is then tranferred to the digitizer and placed in words 2006, 2007 and 2008 of the output buffer. Just before each word of the output buffer is replaced by a new sample, the present word is shipped out and recorded. Thus the starting time for each record is recorded in the previous record.

If the tape speed should vary with the digitizer free-running, the time (as seen from the tape) between data samples will vary. For example: Assume that the tape speed increases by 1% and the digitizer continues to sample at 1 kHz per data channel. If before the speed increase there was 1 msec between each data sample, after the speed increase there would be 1.01 msec between each data sample. This is a small error for <u>one</u> sample. However, the cumulative error after 1000 samples (1 sec of sampling) is 10

Table 5.2 NASA format for BCD time encoding.

Word	bits	
Word 2006	1 - 4	tenths of milliseconds
	5 - 8	one of milliseconds
	9 - 12	tens of milliseconds
	13 - 16	hundreds of milliseconds
Word 2007	1 - 4	ones of seconds
	5 - 7	tens of seconds
	9 - 12	ones of minutes
	13 - 15	tens of minutes
Word 2008	1 - 4	ones of hours
	5 - 6	tens of hours
	7 - 10	ones of days
	11 - 14	tens of days
	15 - 16	hundreds of days

msec.

In order to synchronize the digitizer with the analog tape, the analog tape's reference signal is used to trigger the digitizer. Since the digitizer sampling rate is usually much less than the tape reference frequency, a frequency converter is needed. Figure 5.3 is a block diagram of the digitizer synchronization circuit. A phase locked loop (PLL) is used on the input to prevent loss of trigger signal during signal drop-outs on the tape and to increase the noise tolerance of the system. Additionally, the PLL increases the flexibility of the synchronization circuit. By inserting a ÷M counter into the PLL feedback loop the PLL input frequency can be multiplied by M.

The PLL output is divided by N resulting in a frequency conversion factor of M/N. This circuit can be used whenever the digitizer trigger frequency is a fractional multiple of the tape reference frequency.

5.3 Probe Analysis

Programs WPROBE and SWEEP are used in the analysis of the Langmuir probe data. WPROBE is run with probe data from a NASA digital tape as input. The output is average probe current values.

The probe current values from program WPROBE are then used as input to program SWEEP. Program SWEEP removes the probe current spikes which result from the sweep of probe potential.

A brief description of these programs will be given in this section. Program listings can be found in the appendices.

5.3.1 Program WPROBE. Program WPROBE averages the digital probe samples and converts them to probe current. The averaging period is usually set equal to the spin period. (The spin period can be obtained from the magnetometer trace on the chart record.) This reduces the oscillation

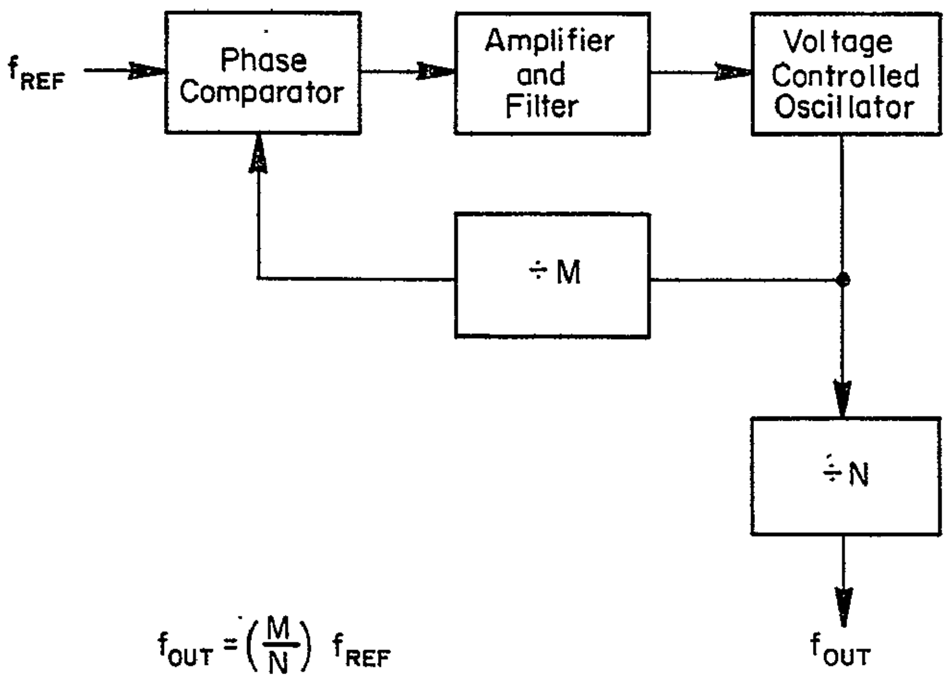


Figure 5.3 Phase-locked loop frequency synthesizer. This circuit develops a frequency (f_{OUT}), required by the digitizer, which is synchronized to the reference frequency (f_{REF}) recorded on the analog magnetic data tape.

effects caused by rocket spin and precession.

The probe current versus time values are output in two forms: printed and punched. The punched output contains only probe current and time data. It is not punched but routed to the Cyber via the fetch queue, where the probe current data eventually are fetched and stored in an indirect access file. This probe current data file can be accessed easily for either interactive or batch processing on the Cyber.

In addition to the probe current versus time values for each output point, the printed output contains values of intermediate conversion variables and altitude information. Also included in the printed output is a listing of the input variables and diagnostic data pertaining to the tape accessing subroutine.

A listing of Program WPROBE along with the necessary JCL for execution on the IBM is given in Appendix I. The input data deck used in processing flight 18.1020 is also listed. Figure 5.4 is a plot of the ascent portion of the probe current output for 18.1020.

The program is run twice: first to determine the calibration voltage and then to process the probe data. The times when the probe is in calibration (see section 4.2.2) can be roughly determined from the chart record. The ascent and descent calibration voltages are used to correct for drift of probe output signal during the flight. The drift is assumed to change linearly during the flight.

The program is documented with comment statements so only a short description of the main processing loop will be given here. Figure 5.5 is a flowchart of the main processing loop. Table 5.3 defines the variables and subroutines used in Figure 5.5.

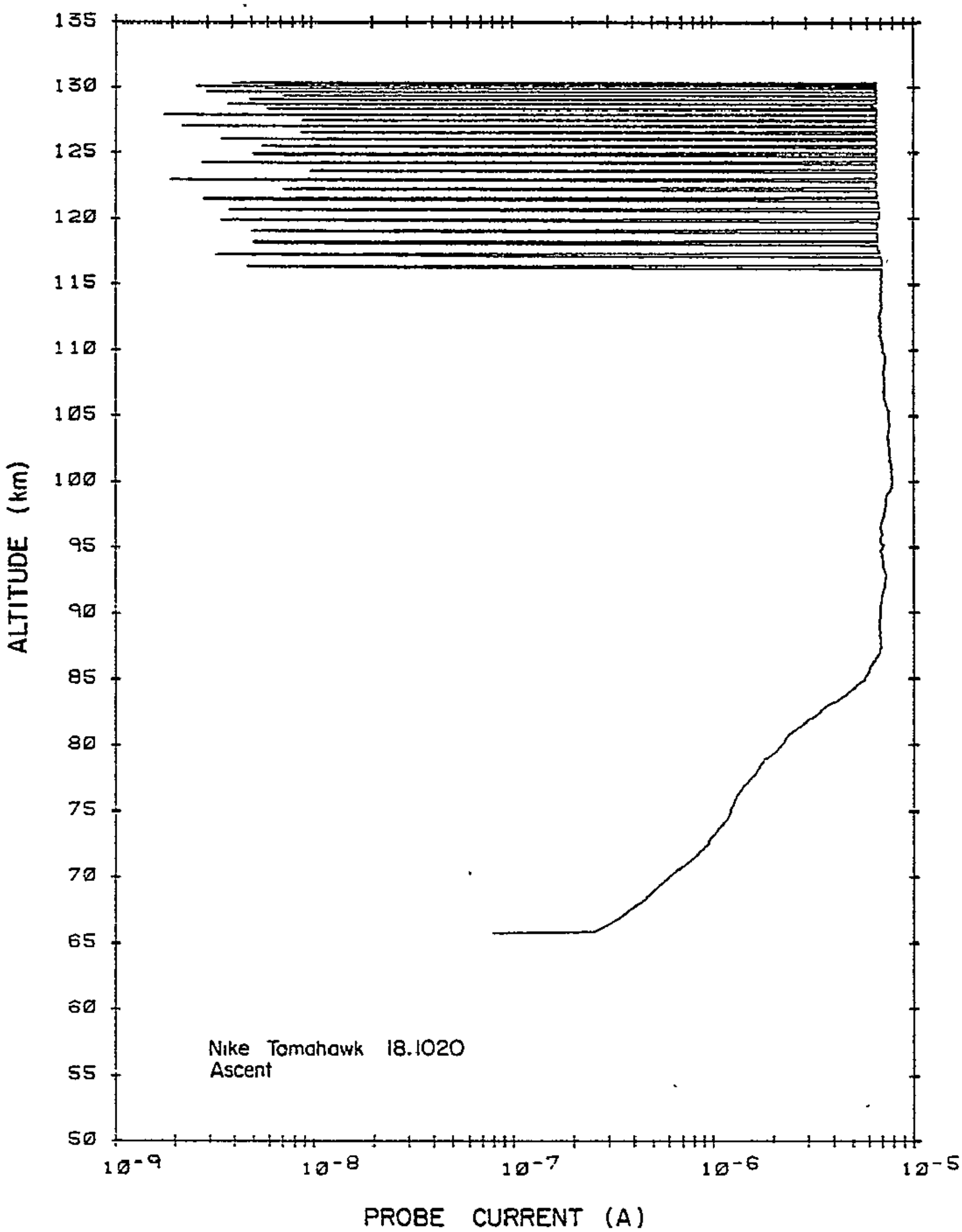


Figure 5.4 Ascent portion of unedited probe current profile for flight 18.1020.

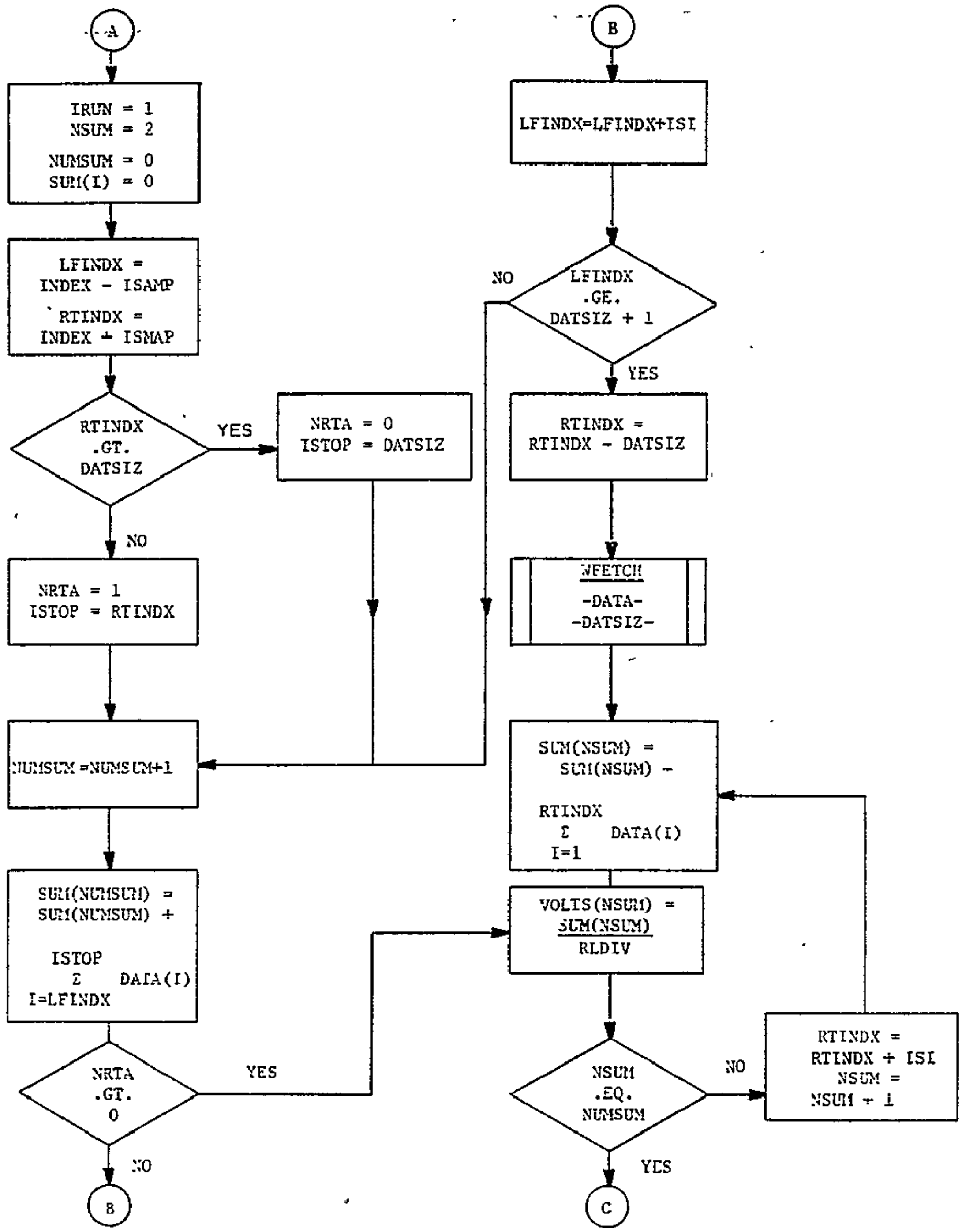


Figure 5.5 Flowchart of the probe current processing loop of program WPROBE. Table 5.3 defines the variables and subroutines found in this figure.

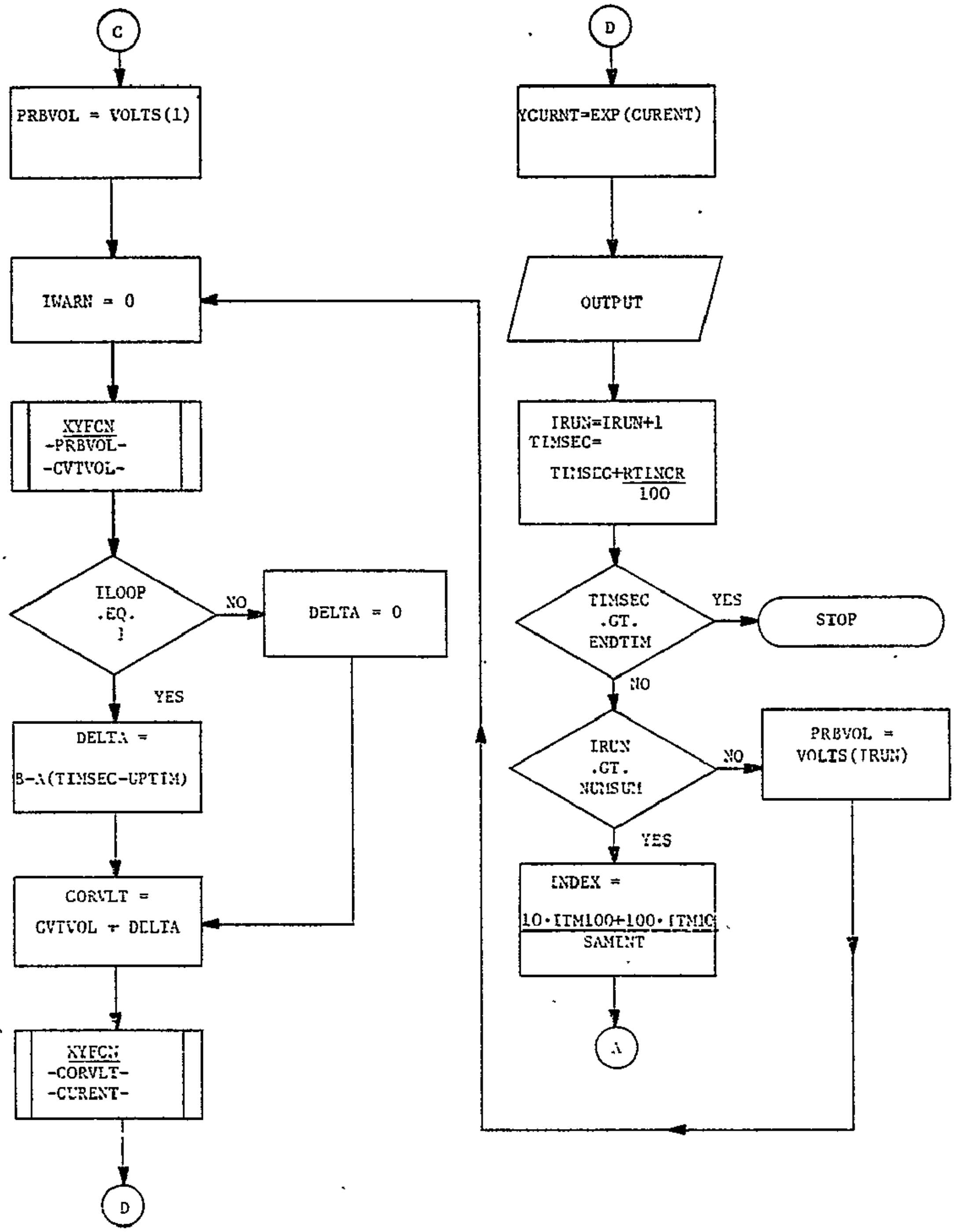


Figure 5.5 (Continued).

Table 5.3 Variables and subroutines in program WPROBE.

IRUN - Number of passes through conversion and output loop.

NSUM - Number of completed sums in present array DATA (these sums spanned two arrays).

NUMSUM - Number of incomplete sums in array DATA (these sums span two arrays).

SUM - Array of integer sums used in computing average data value for each output point.

LFINDX - Subscript of first value in array DATA to be included in average.

INDEX - Subscript in array DATA corresponding to the time of the output point.

ISAMP - Number of data samples per half spin period [(IPERID/2)*(SAMRAT/(1000*NCHANL))].

RTINDX - Subscript of last value in array DATA to be included in average.

DATSIZ - Length of array DATA.

NRTA - Flag: 1 indicates that average can be completed using present array DATA; 0 indicates a span of two arrays.

ISTOP - End of summation in present DATA array.

DATA - Array containing digital values of probe current.

ISI - Number of digital samples between output points.

WFETCH - Subroutine: This subroutine fills array DATA with one second of digital probe current values starting at time TIMST.

VOLTS - Average digital value of probe current.

RLDIV - Number of digital samples in averaging period [(2*ISAMP) + 1].

PRBVOL - Digital value which is to be converted to voltage.

IWARN - Flag: 1 indicates that an interpolation could not be accomplished by XYFCN; 0 indicates interpolation accomplished.

XYFCN - Subroutine: This subroutine performs a linear interpolation.

CVTVOL - Telemetry voltage of average digital value.

ILOOP - Flag: 1 indicates final data run; 0 indicates calibration run.

DELTA - Drift correction voltage [B-A*(TIMSEC-UPTIM)].

Table 5.3 (Cont.)

A - Drift correction slope.

TIMSEC - Time of present output point.

UPTIM - Time of ascent calibration voltage.

B - Drift correction offset.

CORVLT - Corrected telemetry voltage of average digital value (corrected for drift).

CURENT - Log probe current value of average digital value.

YCURNT - Probe current.

RTINCR - Time between output points.

ENDTIM - Last time required for present pass through conversion and output loop.

ITM100 - Hundredths place of TIMSEC.

ITM10 - Tenths place of TIMSEC.

SAMINT - Time between digital samples in milliseconds.

Upon entering the loop of Figure 5.5 array DATA contains one second of probe data (integer value between 0 and 4095). The last value of array DATA corresponds to the time of the first output point. The last ISAMP values of the array are summed.

Subroutine WFETCH is called to fetch the next second of data. The summation for the present output point is completed and then the average is computed. Subroutine XYFCN is called next to convert the average digital value to the corresponding telemetry voltage. For final data runs this voltage is corrected for battery drift. For calibration runs this correction cannot be applied.

The next call of subroutine XYFCN converts the telemetry voltage to the log of the probe current. (Recall that the output of the experiment is the log of the current (see section 4.2.2)). The base of the natural logarithms, (e), is then raised to this power to yield probe current.

This data point is output along with the time, time from launch, altitude, digital value, telemetry voltage and corrected telemetry voltage. The altitude at any time is computed by performing a parabolic interpolation on the 10 second trajectory data.

The time of the next output point is computed. If it is not greater than the stop time, the program loops back to compute the next average.

Eventually a point will be reached where all the data values necessary for an average are not contained in the present DATA array. The incomplete sums of the output points remaining in the present second of data sums are formed before calling WFETCH. Array SUM contains these incomplete sums. NUMSUM contains the number of incomplete sums. After WFETCH has been called these sums are completed and averages are computed. NSUM is the index of these completed sums.

These averages are converted and output one by one before the program loops back.

5.3.2 Program SWEEP. The probe current versus time output from program WPROBE is transferred to the Cyber via the fetch queue. Program SWEEP is run interactively on the Cyber to remove the probe current spikes resulting from the sweeps of probe potential (see Figure 5.4).

Program SWEEP not only removes sweeps, but can be used to plot the probe profile. Trajectory data are needed in order to plot the probe profile. The trajectory file is the same as the trajectory deck input to program WPROBE.

Appendix II contains a listing of program SWEEP. The first nine lines of Appendix II are the Cyber operating system commands necessary to fetch the probe current data from the IBM, save it, and execute program SWEEP. Output profiles of probe current from program SWEEP are plotted in Figures 5.6 and 5.7. These figures are the ascent and descent portions, respectively, of the probe current data from flight 18.1020.

Figure 5.8 is a flowchart of the sweep removal section of program SWEEP. The plotting section of this program is identical to the plotting routine used in program EDPLOT which is described later. The variables named in the flowchart of Figure 5.8 are defined in Table 5.4.

After data are input and variables are intialized, the program begins to look for sweeps. A possible sweep occurs when the ratio of two adjacent probe current values is greater than three. After a possible sweep has been located, the sweep minimum is found. If this minimum is smaller by at least a factor of twenty than the average value expected for that point, a sweep has been located. The index in the probe current array of this sweep minimum is stored. The program then loops back to find the remaining sweeps.

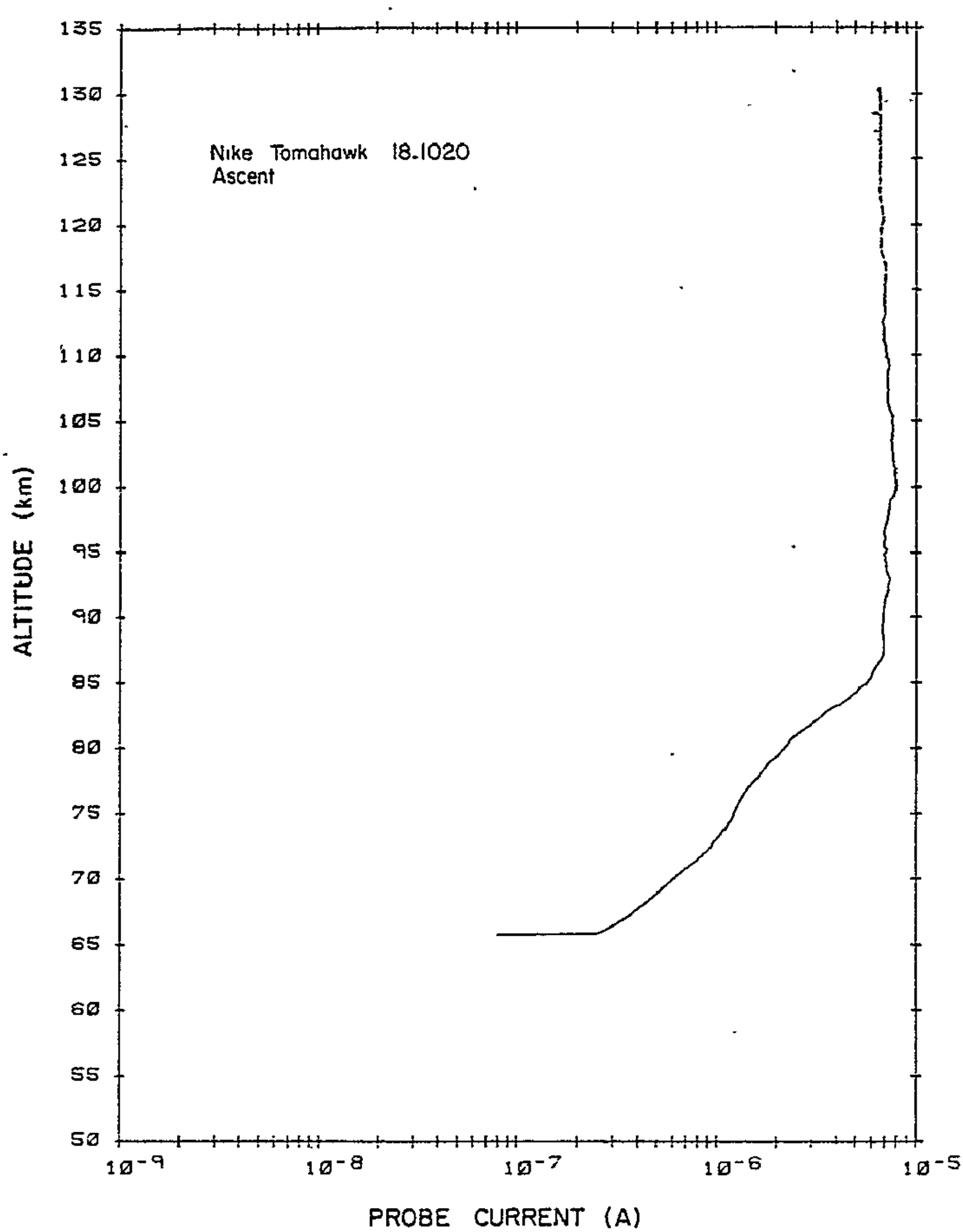


Figure 5.6 Ascent portion of edited probe current profile for flight 18.1020. Program SWEEP has removed the sweeps.

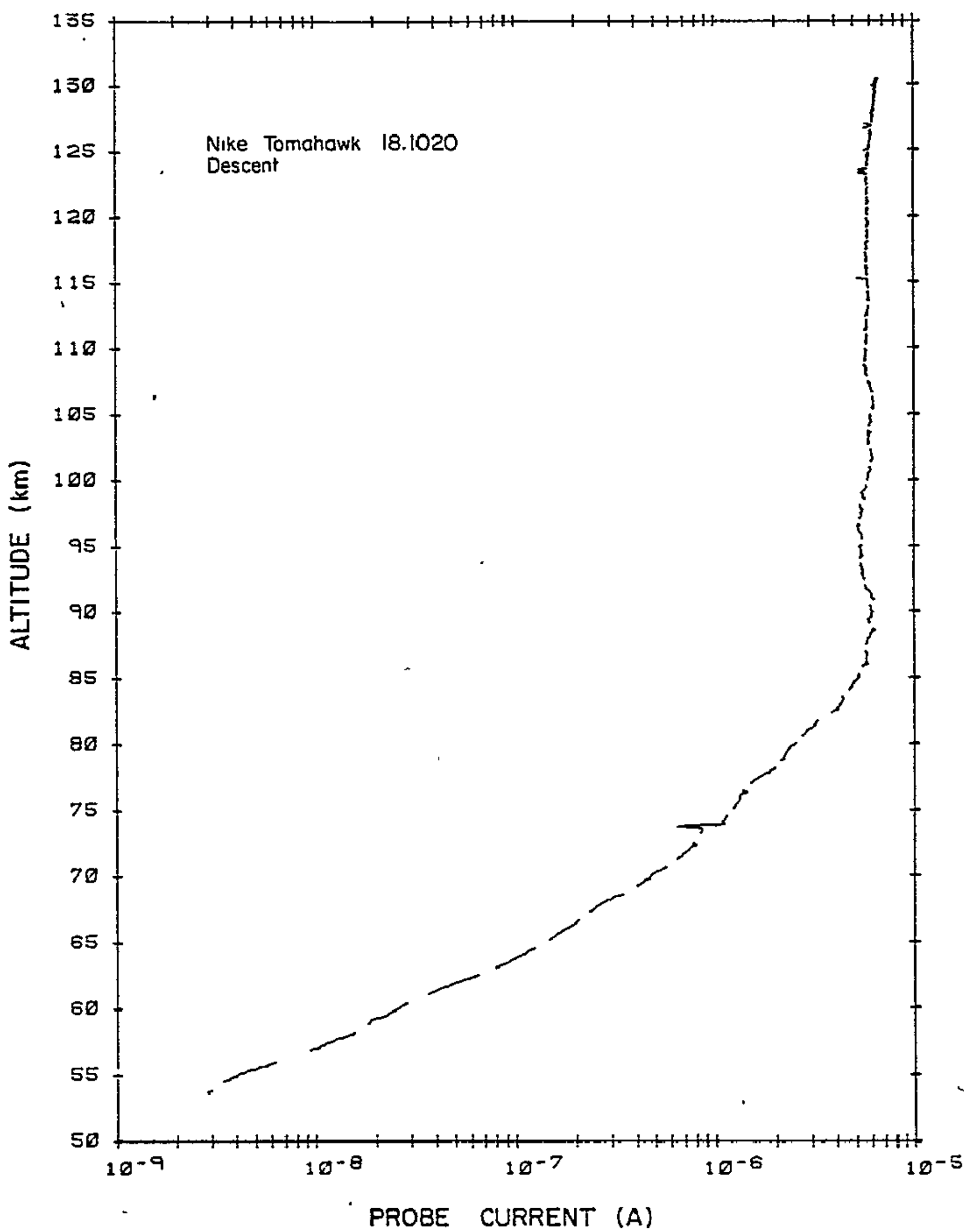


Figure 5.7 Descent portion of edited probe current profile for flight 18.1020. Program SWEEP has removed the sweeps.

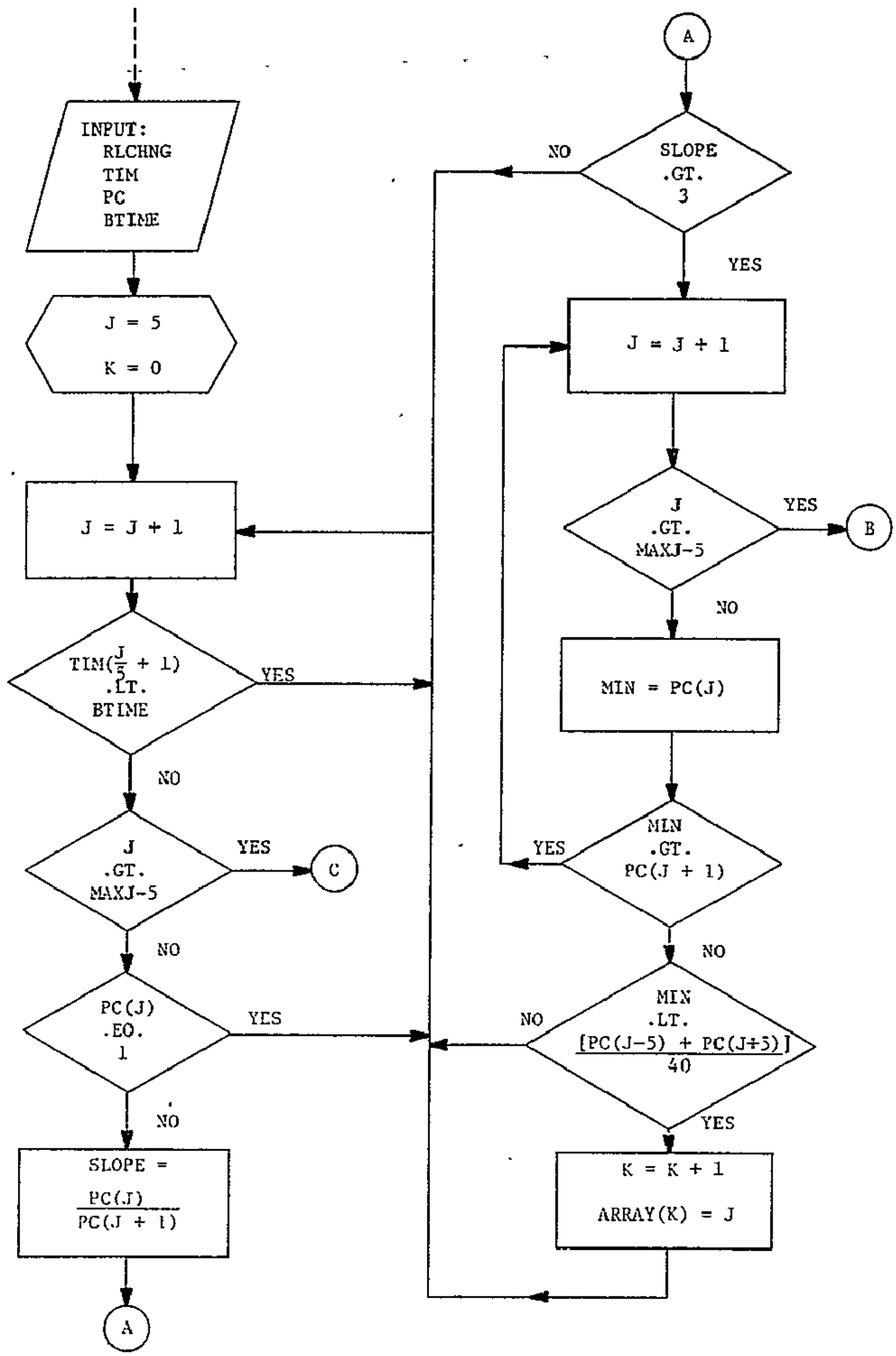


Figure 5.8 Flowchart of the sweep removal section of program SWEEP. Table 5.4 defines the variables found in this figure.

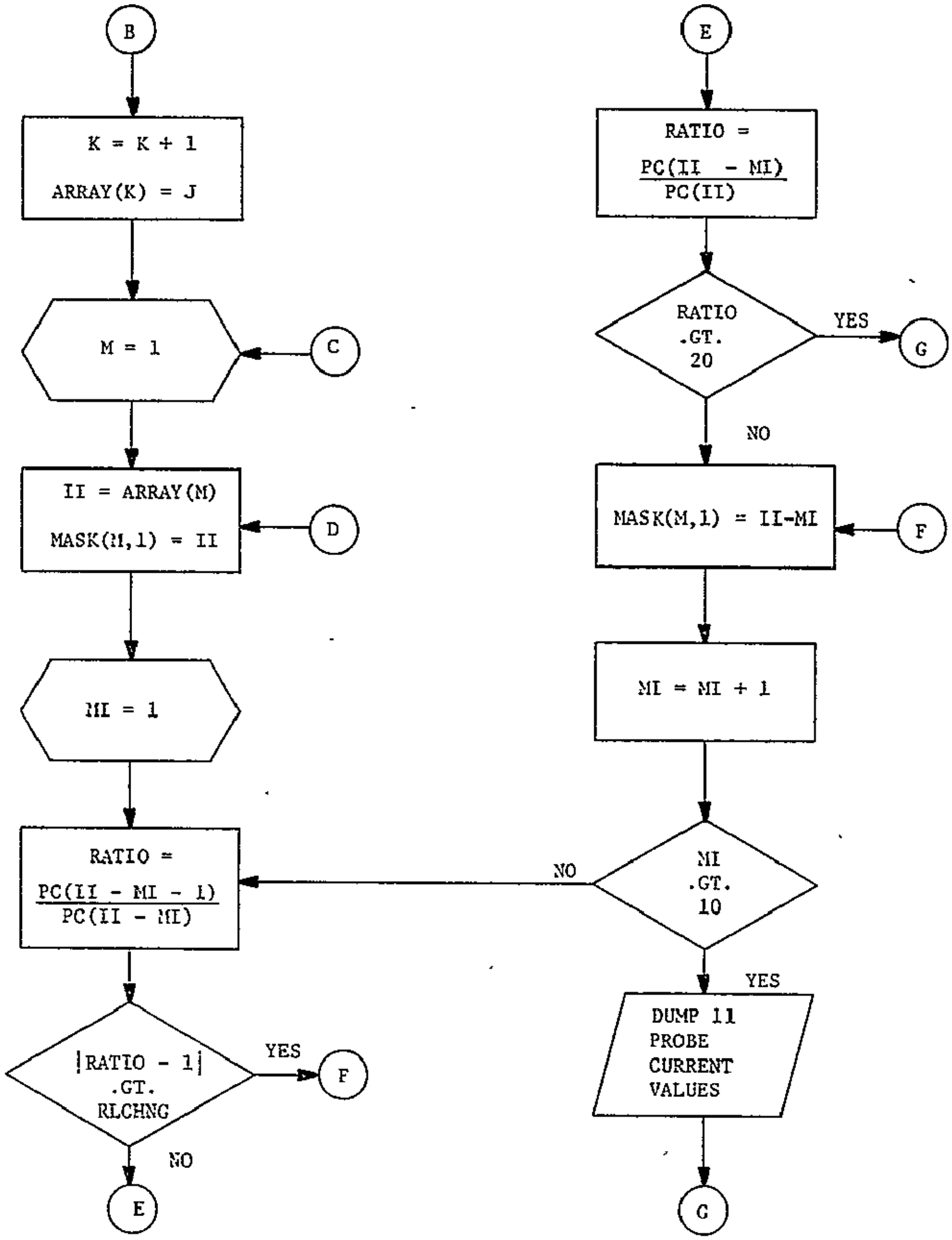


Figure 5.8 (Continued).

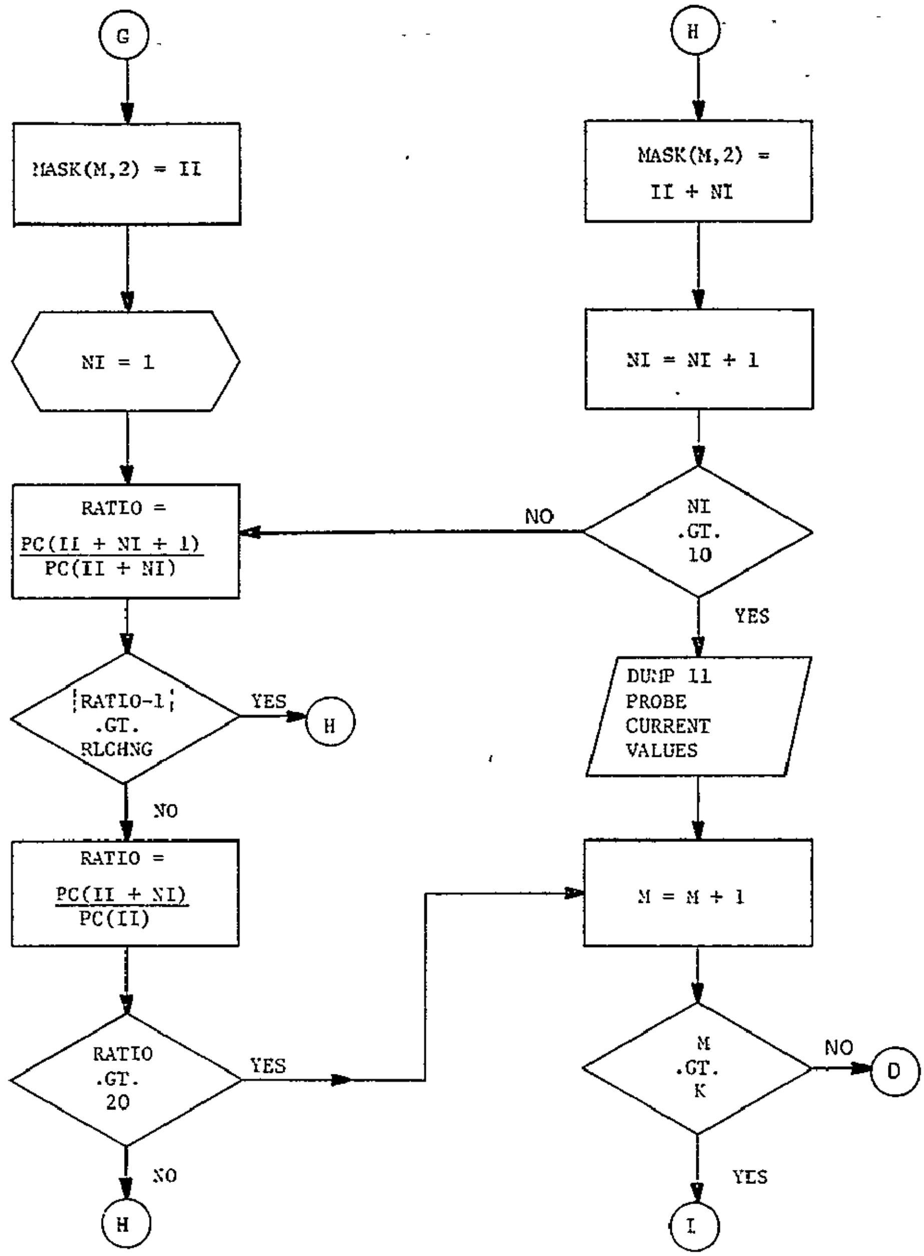


Figure 5.8 (Continued).

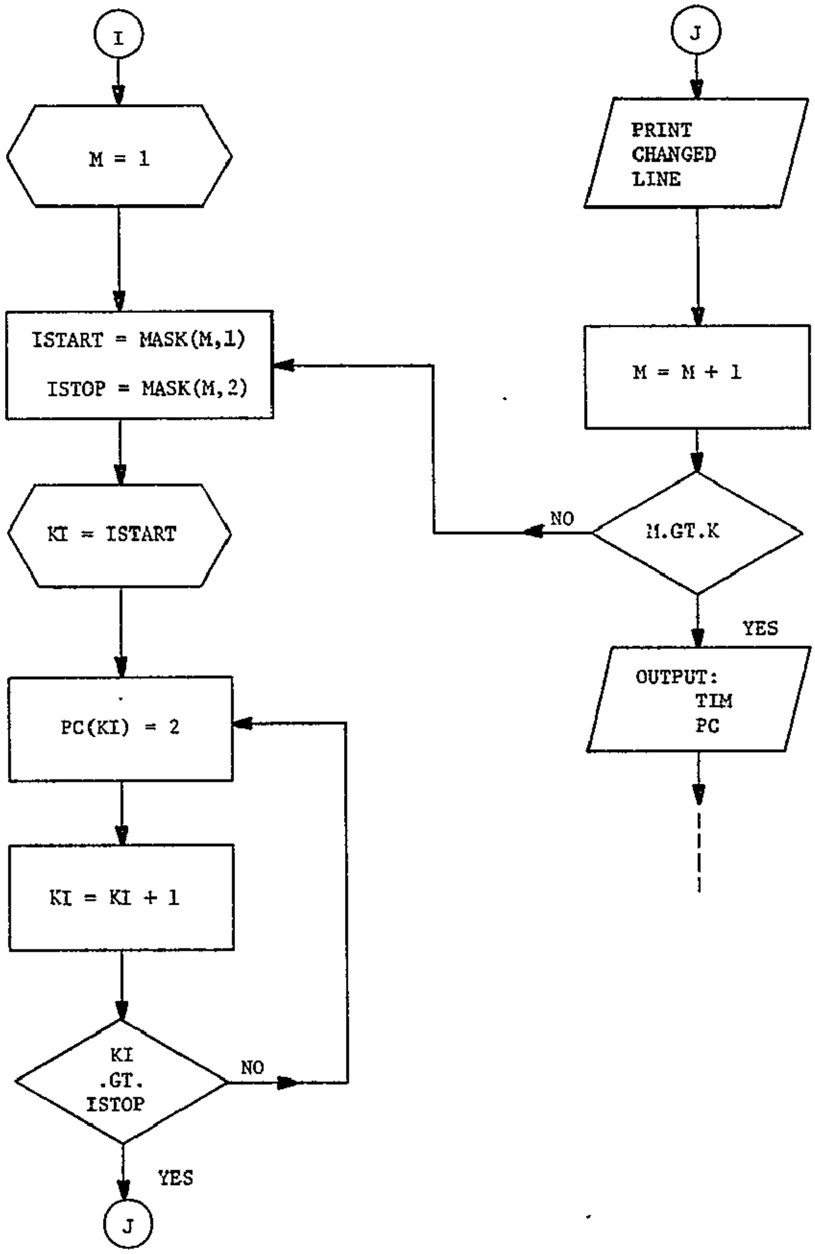


Figure 5.8 (Continued).

Table 5.4. Variables in program SWEEP.

RLCHNG	- Relative change parameter (deviation of the ratio of two probe current values from 1).
TIM	- Array of time values (one time for every five probe current values).
PC	- Array of probe current values.
BTIME	- Time to begin removing sweeps (in seconds from launch).
J	- Index in array PC.
K	- Index in array ARRAY: Number of sweeps.
MAXJ	- Number of probe current values.
SLOPE	- Ratio of two consecutive probe current values (PC(J)/PC(J+1)).
MIN	- Minimum probe current value of current sweep.
ARRAY	- Array containing indices in array PC of sweep minimums.
M	- Do loop index.
II	- Index in array PC of sweep minimum.
MASK	- Two-dimensional array containing beginning sweep index and ending sweep index in array PC.
MI	- Do loop index.
RATIO	- Ratio of two probe current values (e.g., PC(II-MI-1)/PC(II-MI); PC(II-MI)/PC(II)).
NI	- Do loop index.
ISTART	- Index in array PC of the first sweep value.
ISTOP	- Index in array PC of the last sweep value.
KI	- Do loop index.

After all sweep minimums have been identified, the indices in the probe current array of the sweep start and end are located. The start and end of a sweep are defined by the input variable RLCHNG, relative change. Beginning at the sweep minimum and working outward the ratio of two adjacent probe current values is computed. If this ratio is within RLCHNG of unity then a possible start or end of sweep is indicated. For a probe current value to be a start or end of sweep it is necessary to have a ratio greater than twenty to the sweep minimum. The start and end of sweep indices are stored for later processing.

If, after ten probe current values have been examined and a start or end of sweep has not been found, the search is terminated. A start or end of sweep should have been found within ten probe current values of the sweep minimum. These ten probe current values along with the sweep minimum are dumped for examination.

After all sweeps have been identified, the probe current values within each sweep are replaced by a flag value of two. The processed probe current data are output along with diagnostics for each line changed.

The start of sweep and end of sweep values are dependent on the value of RLCHNG entered. If RLCHNG is made too high, too many probe current values will be removed. Conversely, if RLCHNG is made too small not enough probe current values will be removed. For this reason, the choice of the best value for RLCHNG is an iterative process. A routine to plot the probe current profile has been incorporated into program SWEEP to simplify the iteration.

At a Tektronic graphics terminal one can view the probe current profile directly after sweep removal, and immediately reprocess with a different value of RLCHNG if the profile was not acceptable. Since the original probe current file is never overwritten it is always available for viewing as a reference.

Even with the most exacting choice of RLCHNG, not all the profile imperfections can be removed. Some manual editing of the probe current file will be required.

5.4 Propagation Analysis

This section contains a brief description of the methods and programs used in analyzing the Faraday rotation and differential absorption data. Since these forms of the propagation experiment are being phased out and are not expected to be used again, the only computer program which will be diagrammed is the program which matches electron densities to Faraday rotation rates.

Program listings are included in the appendices.

5.4.1 Faraday rotation. There are two fundamental programs used in the computation of Faraday rotation rates. There is one program to match Faraday rotation rates to electron densities.

Faraday rotation rates are derived using a frequency difference scheme (see section 4.3.2). The frequency spectrum of the receiver output is found digitally using the Fast Fourier Transform (FFT). The accuracy of the FFT is dependent on the sampling rate. Therefore, a sampling rate of 25 kHz is used in digitizing the Faraday rotation data.

The first program to be run is TIMLIS. This program lists the time code which is encoded at the end of each record. For the FFT to maintain accuracy it is required that no skipping of time occur between data records (see section 5.1) (i.e., each data record must have the same sampling rate). Program TIMLIS lists any discrepancies of the record time lengths from their nominal value.

The record times are used to locate the starting data point for program FFTR.

Program FFTR performs the FFT. The spectrums of the receiver output, the 500 Hz reference signal and the magnetometer channels need to be computed. The fundamental frequency from each of these channels is used to determine the Faraday rotation rate.

The Faraday rotation rate and the rocket flight parameters are input to program FR2NE to find the electron density.

5.4.1.1 Program TIMLIS. Appendix III lists program TIMLIS. The first 19 lines of the appendix are a Cyber batch job which will run program TIMLIS.

The lines of the batch job following /EOR are the input for the program. The first input card is the expected record length in milliseconds times ten. In this example 0800 corresponds to a record length of 0.08 seconds (80 msec).

The second input card contains the record processing information. The first integer is the number of the first record to examine. The second integer is the number of consecutive records to examine. The last integer is the output selection number. A 0 will print record times for those records whose lengths are different from the expected lengths entered on the first input card. A 1 will print all record times.

In this example the times of records 1 through 1500 will be printed. Additional record processing cards may be included.

Program TIMLIS also outputs the header information for records of length less than 45 words.

The program flow is straightforward: Data cards are read in; heading is printed; records are examined; next data card is read in; etc. There are two subroutines which may be baffling. One is subroutine BACKUP, the other is subroutine GBYTES.

Subroutine GBYTES unpacks one array into another. In subroutine TPGET it is used to convert the 16-bit IBM words into 60-bit CDC words. Each tape record is read into array BUFFER starting with the first location and transferring bit to bit. This means that a 2008 word IBM record (tape) will be stored in 536 CDC words with 32 bits left over ((536 x 60) - (2008 x 16) = 32). Subroutine GBYTES walks through array BUFFER taking each consecutive 16-bit group and transferring it to a word of array ARRAY beginning with location M. Subroutine GBYTES can be used to perform the word conversion only when the words contain character or positive integer data.

In the main processing loop subroutine GBYTES is used to split the 16-bit IBM words (which are now stored right justified in a 60-bit CDC word) into two 8-bit bytes and to store them in array PARRAY. Each element of PARRAY contains one EBCDIC character. Compass function CONETD is called to convert each EBCDIC character into a DISPLAY character.

Subroutine BACKUP is used to backspace one record on the IBM data tape. CDC fortran inline function BACKSPACE will not work with IBM created tapes because of the different record structure. This makes it necessary to backspace "manually". Subroutine BACKUP and integer function FETADR are used for this purpose.

5.4.1.2 <u>Program FFTR</u>. Program FFTR is listed in Appendix IV along with the JCL cards and input data cards necessary to run the program on the IBM. Each data card contains four input parameters: the number of the record in which the first data point is located, the number of the frame in which the first data point is located, the position in the frame of the first data point (i.e., channel number) and the number of points to be used in spectral analysis (usually one second of data).

The input data cards in Appendix IV represent the analysis of 34 seconds of data from channel 4 for flight 18.1020. The location of the first data point to be processed can be interpolated from the output of program TIMLIS. For flight 18.1020. the spectral analysis was done from T + 50 to T + 83 seconds. To center the analysis on T + 50, the first data point must correspond to T + 49.5 or 1652:49.500. This time falls in record 995 (from program TIMLIS) which has a start time of 1652.49.482. Each record contains 400 frames ([2008-8]/5) and is 80 msec long (400/5000). So 1652:49.500 will fall in frame number 90 ([[49.500-49.482]/0.08] x 400). The other input cards are calculated in a similar manner.

If each record is 80 msec long (with no time slippage) then the computation of the additional data cards is simplified because patterns are formed in the record number differences and the frame numbers.

The output from program FFTR is a listing of the spectral peaks between 0 and 69 Hz and between 470 and 558 Hz along with a spectral plot for each second. The plot aids in identifying the spectral peaks and determining the spectral noise level (see Figure 5.9).

Two special algorithms are used in program FFTR. One to apply a Gaussian window to the data before transforming and the other to calculate the spectral peaks. The Gaussian window is given by

$$w(1) = \exp(-\delta^2 /2) - C_f \qquad (5.1)$$

where

$$\delta = 3(1.0002 - \frac{1}{2500}); \; 1 = 1.\ 2,\ \ldots,\ 2500$$

and

$$C_f = 0.01111$$

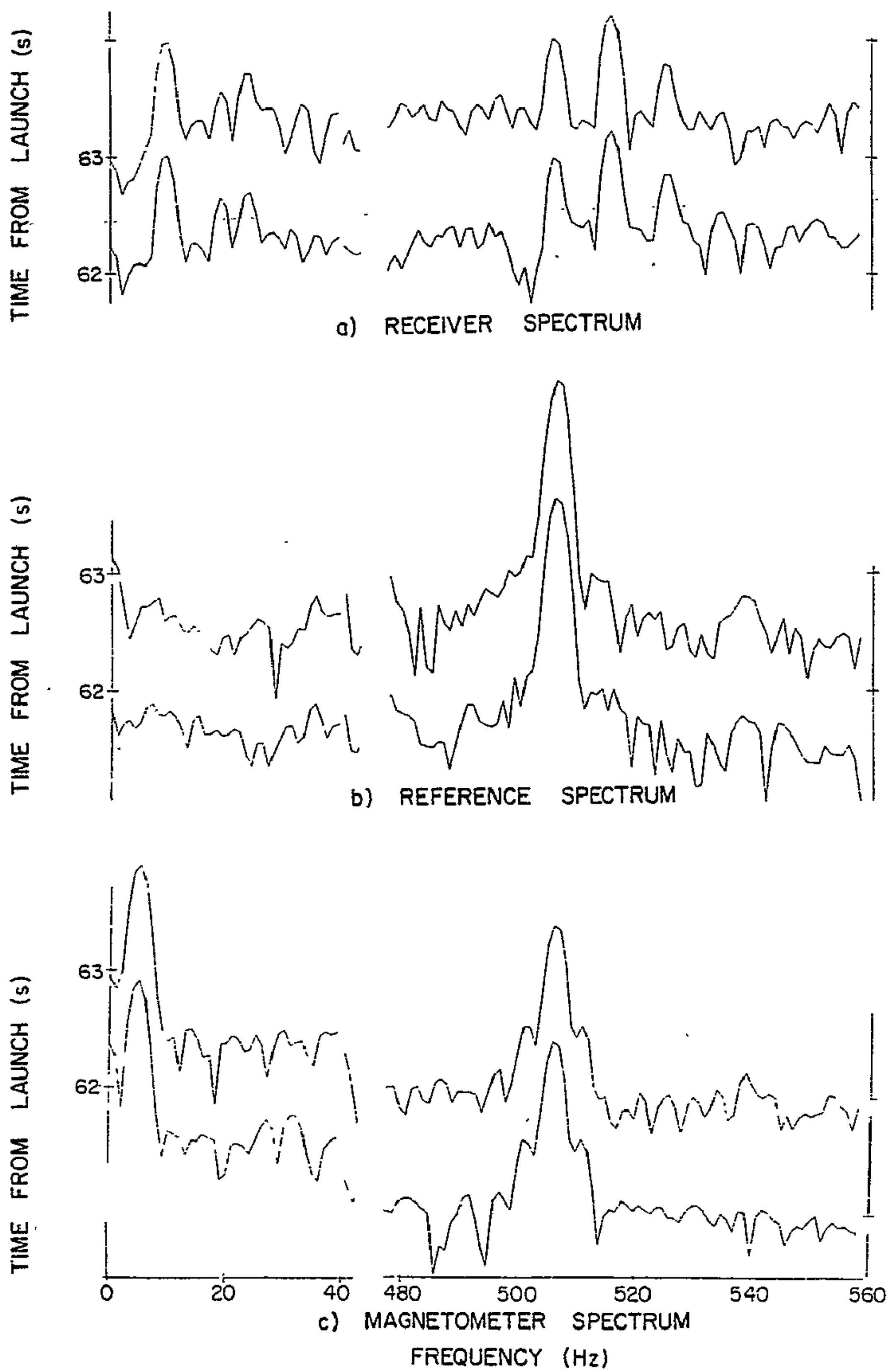


Figure 5.9 Frequency spectra of data tape channels 3, 4 and 5 at T+62 and T+63 seconds from the 5 MHz Faraday rotation experiment for flight 18.1020. Receiver spectra (channel 3) are shown in (a), reference spectra (channel 4) in (b) and magnetometer spectra (channel 5) in (c).

The spectral peak is computed using the algorithm

$$f_s = f_2 + \frac{1}{2}[(1 - R)/(1 + R)] \tag{5.2}$$

where

$$R = \ln(X_2/X_3)/\ln(X_2/X_1)$$

With X_1, X_2 and X_3 being the magnitudes of the spectral densities at consecutive component frequencies f_1, f_2 and f_3 (i.e., $f_1 = (k-1)$, $f_2 = k$ and $f_3 = (k+1)$ where k is the index).

The data windowing is discussed in detail in FILLINGER ET AL. (1976) and GILCHRIST AND SMITH (1979). The algorithm for determining spectral peaks is derived and discussed in EDWARDS (1975).

5.4.1.3 Program FR2NE. Faraday rotation rates are matched with electron density values via program FR2NE. Appendix V lists program FR2NE. The first 40 lines of the appendix are a CDC Cyber batch job used to run program FR2NE. Following the /EOR card is the input data. The first line of input contains the flight number, the date of launch, the frequency of the experiment and the collision frequency model parameter.

This program uses the Sen-Wyller generalized magnetoionic equations to compute Faraday rotation rates. The electron density is iterated until the experimental Faraday rotation rate is matched. Because the Sen-Wyller equations are generalized to include the ionized particles' collisions with neutral particles a collision frequency must be included in the calculations. This frequency can be either computed or modeled. If differential absorption data are combined with Faraday rotation data, then the collision frequency can be computed along with the electron density (i.e., two equations, two unknowns). When Faraday rotation data is used alone, then a model must be used for the collision frequency parameter.

In program FR2NE the collision frequency parameter is modeled as a constant times the atmospheric pressure (SMITH ET AL., 1978).

The remaining input data cards contain rocket location information along with the experimental Faraday rotation rates. The rocket location information consists of the look azimuth and the look elevation (the azimuth and elevation of the rocket referenced to the location of the transmitting antennas used for the experiment), the height above sea level, the total velocity (velocity in the direction of rocket motion), and the rocket latitude and longitude (referenced to an oblate earth). The above quantities are all available as the trajectory data supplied by NASA. One needs to supply NASA with the location of the transmitting antennas so that the look azimuth and look elevation can be included.

Figure 5.9 shows the frequency spectrums associated with the 18.1020 5 MHz propagation experiment. Figure 5.9c is a spectrum of the magnetometer channel for T+62 and T+63 seconds. Figures 5.9b and a are spectra of the reference and receiver signals, respectively, for T+62 and T+63 seconds. These six curves are edited output from program FFTR.

The experimental Faraday rotation rate f_F (in Hz) is given by

$$f_F = \frac{f_{receiver} - f_{reference} - 2f_{spin}}{2} \tag{5.3}$$

Each of these frequencies is listed in a printed output from program FFTR. An estimate of a frequency can be made from the spectral plot, then the actual frequency can be located in the printed output. The spin and reference frequencies (Figures 5.9c and b) correspond to the greatest spectral peaks. Note that there are three spectral peaks in the 500 Hz vicinity of Figure 5.9a. The upper peak represents the extraordinary component of the signal; the lower peak represents the ordinary component of the signal; and the center peak represents the combined signal. As the extraordinary wave is absorbed at the higher electron densities, the upper peak diminishes. When the extraordinary wave is completely absorbed only the lower peak, representing the ordinary wave, remains. Eventually, the ordinary wave will be absorbed also.

Listed in Tables 5.5 and 5.6 are the frequency components used to calculate the Faraday rotation rates for 18.1020. Table 5.5 contains Faraday rotation data for the 2 MHz propagation experiment. Table 5.6 contains Faraday rotation data for the 5 MHz propagation experiment. The Faraday rotation rate is converted to degrees/sec by multiplying by 360.

The output from program FR2NE consists of a header and a body. The header contains the definitions of the variables listed in the body. Also contained in the header are the input data quantities.

The body contains the electron density and computed Faraday rotation rate along with other variables used in the computation of the Faraday rotation rate. The experimental Faraday rotation rate is matched through an iterative process. Every second iteration step is output. The iteration is performed forty times resulting in twenty output steps for each experimental Faraday rotation rate.

An abbreviated flowchart of program FR2NE is shown in Figure 5.10. Table 5.7 is a listing of the variables and subroutines found in Figure 5.10.

All subroutines are listed in Appendix V. Subroutine PRESSR and its called subroutines (listed after subroutine PRESSR in the appendix) are pre-compiled and placed in the subroutine library BLIB. This reduces compilation time.

Table 5.5 18.1020 2 MHz Faraday rotation.

Time from Launch (sec)	f_{spin} (Hz)	$f_{reference}$ (Hz)	$f_{receiver}$ (Hz)	$f_{Faraday}$* (Hz)	$f_{Faraday}$ (deg/sec)	Electron Density m^{-3}
50	8.83918	491.31445	508.97803	-0.00739	-2.6604	unmatchable
51	8.70103	491.31567	508.68701	-0.01536	-5.5296	unmatchable
52	8.61405	491.31445	508.54761	0.00253	0.9108	1.444×10^9
53	8.60254	491.31494	508.51904	-0.00049	-0.1764	unmatchable
54	8.59799	491.31421	508.53052	0.01016	3.6594	4.324×10^9
55	8.05849	491.31445	502.38672	-2.52236	-188.0478	1.359×10^{11}
56	4.77360	491.31372	500.85010	-0.00541	-1.9476	unmatchable
57	4.77857	491.31421	500.84448	-0.01344	-4.8366	unmatchable
58	4.78039	491.31348	500.85718	-0.00854	-3.0744	unmatchable
59	4.78203	491.31421	500.87964	0.00068	0.2466	2.329×10^8
60	4.78207	491.31372	500.85083	-0.01352	-4.8654	unmatchable
61	4.78293	491.31494	500.89844	0.00882	3.1752	3.989×10^9
62	4.78172	491.31372	500.85986	-0.00865	-3.1140	unmatchable
63	4.78208	491.31274	500.86719	-0.00486	-1.7478	9.436×10^9
64	4.78115	491.31445	500.81714	-0.02980	-10.7298	2.340×10^{10}

$$*f_{Faraday} = \frac{f_{receiver} - f_{reference} - 2f_{spin}}{2}$$

Table 5.6 18.1020 5 MHz Faraday rotation.

Time From Launch (sec)	f_{spin} (Hz)	$f_{reference}$ (Hz)	$f_{receiver}$ (Hz)	$f_{Faraday}$* (Hz)	$f_{Faraday}$ (deg/sec)	Electron Density m^{-3}
50	8.83918	506.29346	523.95068	-0.01057	-3.8052	unmatchable
51	8.70103	506.29395	523.65430	-0.02086	-7.5078	unmatchable
52	8.61405	506.29297	523.52905	0.00399	1.4364	9.322×10^{9}
53	8.60254	506.29419	523.50415	0.00244	0.8784	6.633×10^{9}
54	8.59799	506.29199	523.49707	0.00455	1.6380	1.654×10^{10}
55	8.05849	506.29248	522.82666	0.20860	75.0960	unmatchable
56	4.77360	506.29102	515.83105	-0.00358	-1.2906	8.680×10^{10}
57	4.77857	506.29150	515.82544	-0.01160	-4.1760	1.770×10^{11}
58	4.78039	506.29102	515.84033	-0.00574	-2.0646	2.294×10^{11}
59	4.78203	506.29102	515.84595	-0.00456	-1.6434	2.743×10^{11}
60	4.78207	506.29053	515.85938	0.00236	0.8478	1.226×10^{9}
61	4.78293	506.28931	515.86255	0.00369	1.3284	1.428×10^{9}
62	4.78172	506.28882	515.85913	0.00344	1.2366	1.028×10^{9}
63	4.78208	506.28955	515.86230	0.00430	1.5462	1.047×10^{9}
64	4.78115	506.28955	515.85986	0.00400	1.4418	8.388×10^{8}
65	4.66741	506.29077	515.69751	0.03596	12.9456	6.543×10^{9}
66	4.62454	506.28906	515.60889	0.03538	12.7350	5.713×10^{9}
67	4.62472	506.29053	515.59619	0.02811	10.1196	4.146×10^{9}
68	4.62558	506.28979	515.68164	0.07034	25.3242	9.602×10^{9}
69	4.62577	506.29028	515.96436	0.21127	76.0572	2.659×10^{10}
70	4.62597	506.28979	515.50244	-0.01964	-7.0722	unmatchable

$$*f_{Faraday} = \frac{f_{receiver} - f_{reference} - 2f_{spin}}{2}$$

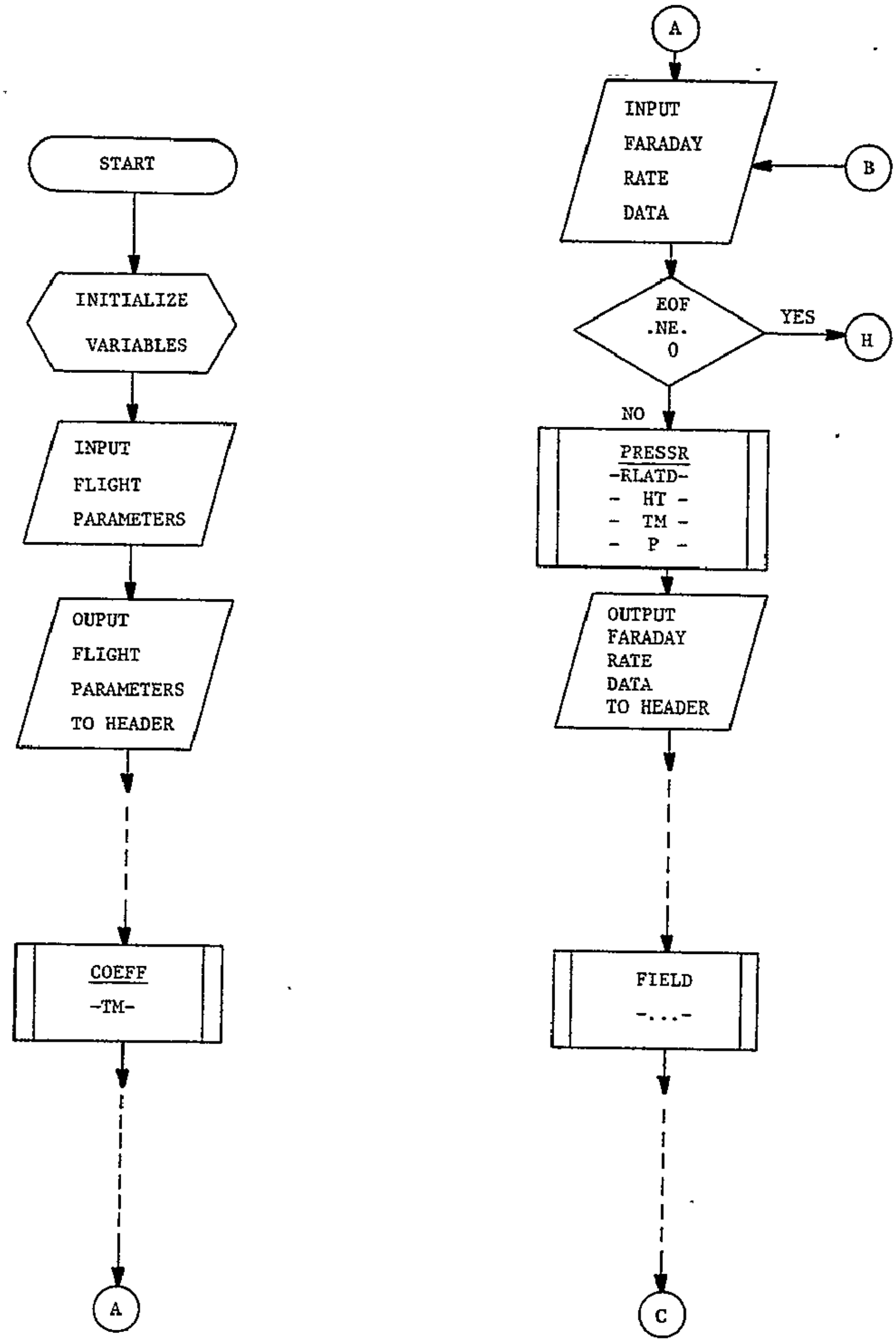


Figure 5.10 Flowchart of the processing loop which matches electron density to Faraday rotation rates in program FR2NE. Table 5.7 defines the variables and subroutines found in this figure.

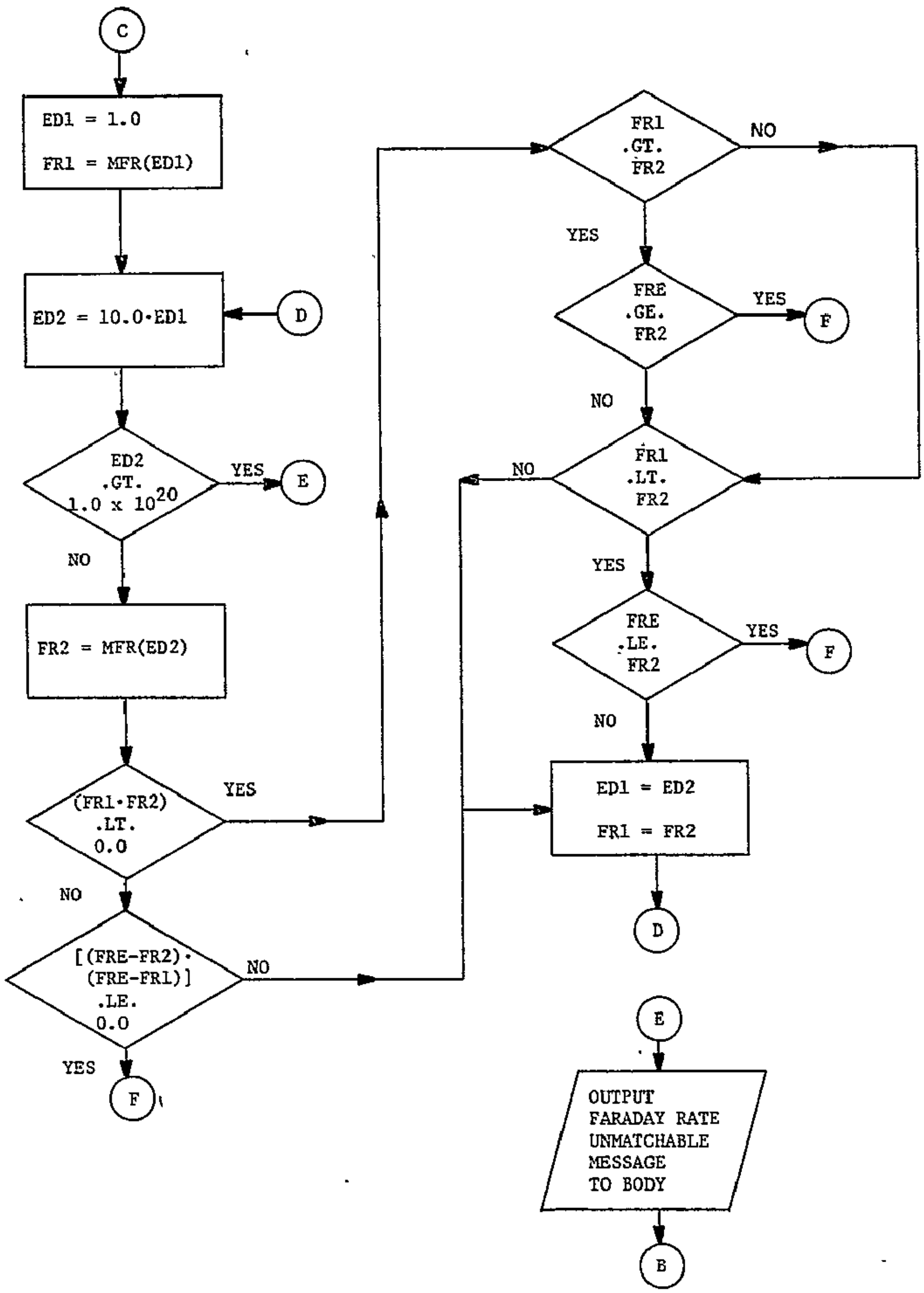


Figure 5.10 (Continued).

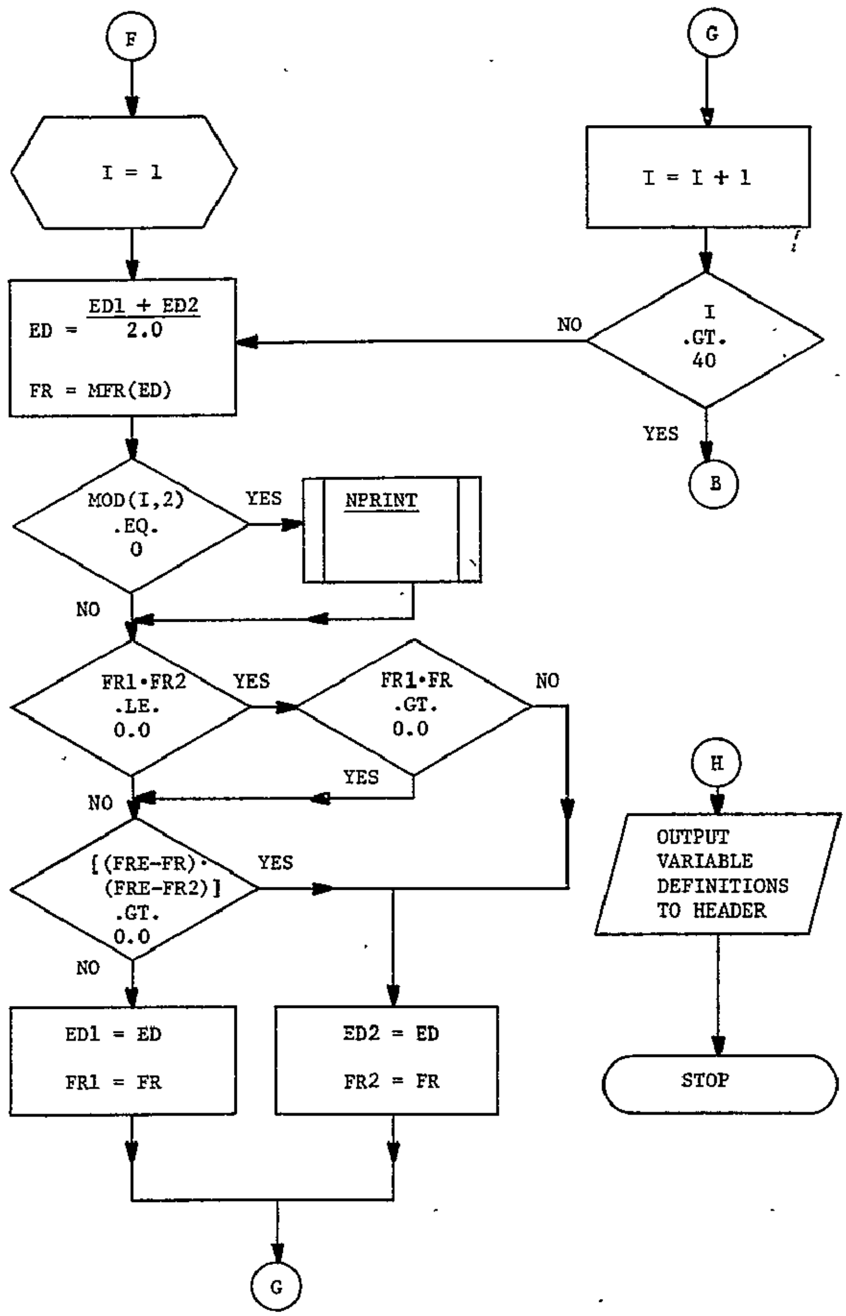


Figure 5.10 (Continued).

Table 5.7 Variables and subroutines in program FR2NE.*

COEFF	- Subroutine: This subroutine computes magnetic field coefficients.
TM	- Date of flight.
EOF	- Flag: 0 indicates read operation successfully completed; other value indicates end of file.
PRESSR	- Subroutine: This subroutine computes atmospheric pressure at a particular latitude and altitude for a time of year.
RLATD	- Rocket latitude.
HT	- Rocket altitude.
P	- Atmospheric pressure at rocket's location.
FIELD	- Subroutine: This subroutine calculates the magnetic field at the rocket's location.
ED1	- Electron density at the lower end of the electron density interval.
FR1 (DA1)	- Theoretical Faraday rotation rate computed at electron density ED1.
MFR (MDA)	- Subroutine: This function subroutine computes the Faraday rotation rate as a function of electron density.
ED2	- Electron density at the upper end of the electron density interval.
FR2 (DA2)	- Theoretical Faraday rotation rate computed at electron density ED2.
FRE (DA2)	- Experimental Faraday rotation rate.
I	- Do loop index.
ED	- Electron density at the center of the electron density interval.
FR (DA)	- Theoretical Faraday rotation rate computed at electron density ED.
MOD	- Subroutine: This function subroutine calculates the modolus.
NPRNT	- Subroutine: This subroutine prints out electron density and other variables used in calculating the Faraday rotation rate.

*Variables in parentheses are the program DA2NE equivalent variables in program FR2NE.

Subroutine PRESSR contains atmospheric pressure data from the CIRA 1972 reference atmosphere. Pressure data for the altitude range 25 km to 500 km are stored in subroutine PRESSR. Data are stored according to latitude, month and altitude. A logarithmic interpolation of pressure is applied using the two altitudes bracketing the entered altitude.

Subroutines COEFF and FIELD calculate the geomagnetic field at the rocket's location. A description of these programs can be found in CAIN ET AL., 1967.

Function subroutine MFR computes the Faraday rotation rate as a function of electron density. It simplifies the calling of subroutine SENWYL. Subroutine SENWYL realizes the generalized magnetoionic equations of SEN AND WYLLER (1960). MECHTLY ET AL. (1970) describes the implementation of the Sen-Wyller equations in this subroutine.

Subroutine NPRNT prints out the values of many of the variables used in subroutine SENWYL. Some variables which are included are: electron density, Faraday rotation rate, differential absorption rate and geomagnetic field components.

The iteration to match an experimental Faraday rotation rate is performed by repeatedly halving the electron density interval and then redefining the interval based on a comparison of the experimental Faraday rotation rate with the computed Faraday rotation rates. The electron density interval is originally set to the powers of ten which result in computed Faraday rotation rates which bracket the experimental Faraday rotation rate. This initialization is performed by the steps on page 2 of the flowchart in Figure 5.10.

Page 3 of Figure 5.10 diagrams the iteration scheme to match the experimental Faraday rotation rate.

The electron density value for a corresponding Faraday rotation rate is chosen by comparing the experimental and computed Faraday rotation rates in the listed output and selecting the electron density which resulted in equal Faraday rotation rates. At lower altitudes an experimental Faraday rotation rate can sometimes be unmatchable due to a discontinuity in the Faraday rotation rate versus electron density curve (see section 6.2.2).

Tables 5.5 and 5.6 contain a listing of the electron densities which result in computed Faraday rotation rates equal to the experimental Faraday rotation rates for flight 18.1020. These electron densities are plotted versus altitude in Figure 5.11.

5.4.2 Differential absorption. Four computer programs are required to reduce the differential absorption data to electron densities. The first program runs on the IBM while the remaining three can be run as batch jobs on the Cyber.

Program DACAL is the first program to be executed. This program is used to determine the calibration levels of the ordinary and extraordinary attenuators (see section 4.3.2, Figure 4.8). The attenuator calibration levels are needed to convert the digitized data to decibel power levels.

Next program CONVWAL is run. Program CONVWAL converts portions of the IBM format data type to a Cyber format data file. Succeeding data anlaysis can be performed using the Cyber computer.

Programs DAMED and DAAVG perform the data reduction. Program DAMED finds the median digital value of a one second interval and, using the calibration levels, converts this digital value to a decibel power level. Program DAAVG computes the average digital value over a one second interval and converts this average value to a decibel power level. In the analysis of the eclipse data program DAMED was used.

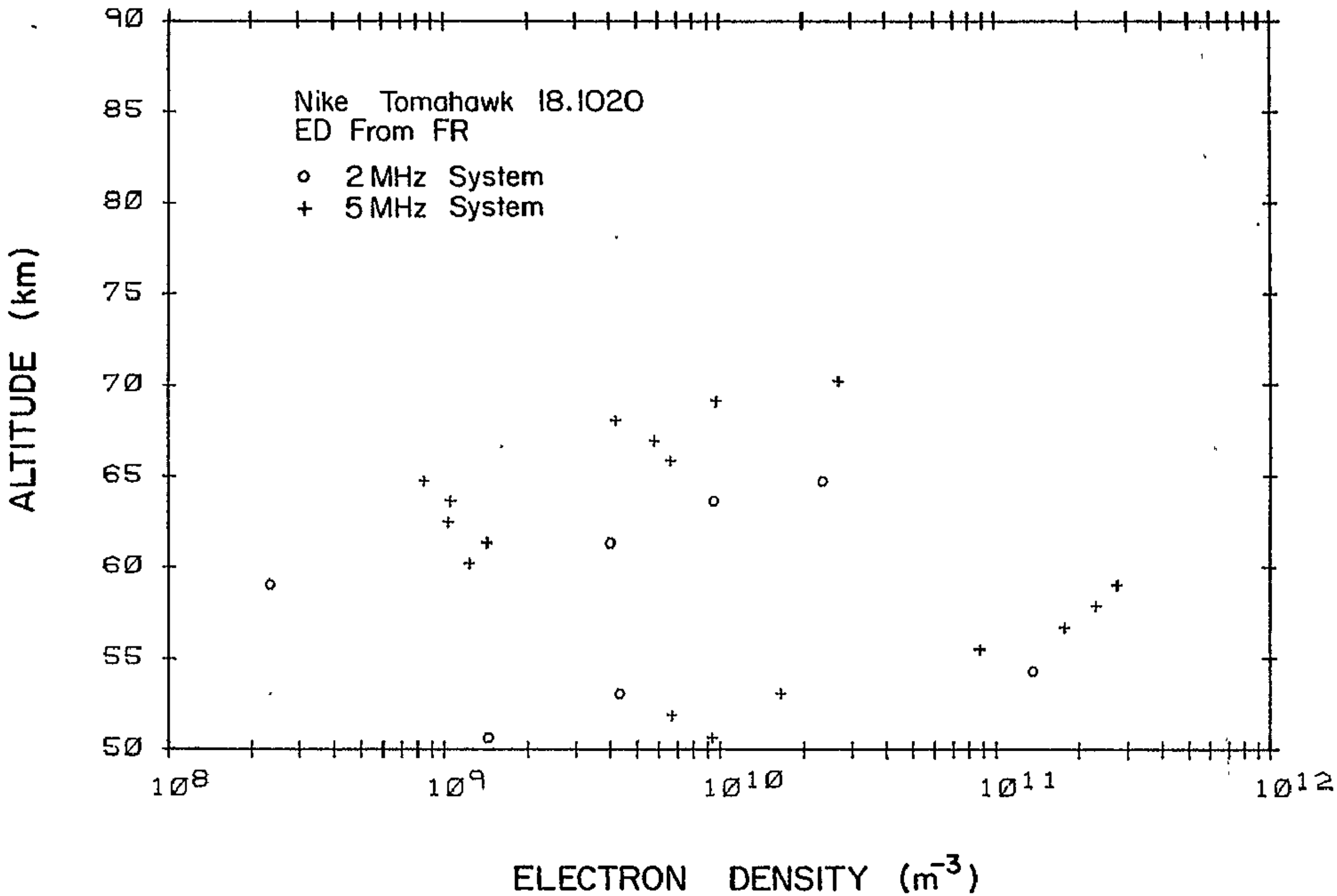


Figure 5.11 Plot of electron densities derived from Faraday rotation rates for flight 18.1020. The circles represent the 2 MHz experimental results while the crosses represent the 5 MHz experimental results.

From the output of program DAMED relative extraordinary decibel power levels are computed. These power levels are plotted and smoothed by drawing a smooth curve through the data points. The differential absorption rate is found by computing the slope of the curve. The experimental differential absorption rates are matched to electron densities via program DA2NE.

The above programs are listed in the appendices.

5.4.2.1 Program DACAL. Appendix VI contains a listing of program DACAL. The JCL required for compilation and execution on the IBM is also included in the appendix. If the program is submitted as listed in the appendix one minute and fifteen seconds from the differential absorption data tape for flight 18.1020 will be analyzed.

There are only two input parameters to program DACAL: the times (in UT) to begin and end processing. These times can be determined either by listening to the voice track of the analog tape or from reading the countdown checklist for the rocket flight. Prior to launch the ordinary and extraordinary attenuators are set and held for five seconds at 10 dB increments from -50 to 0 dB. This records six attenuator calibration levels on the NASA analog tape. The performance of each calibration step is recorded on the voice channel of the NASA analog tape. By replaying the analog tape and monitoring the voice and time channels the times when the calibration steps were performed can be determined. The time interval during which all the calibration steps were performed is used as input to program DACAL.

Program DACAL computes an average digital value for each consecutive group of fifty data points. These average values are printed for each of the five data channels. Record time is printed once a second to simplify locating the calibration levels. By glancing at the average digital values for a calibration step an estimate of the calibration level can be made. An exact value can be obtained by averaging the averages.

Program flow is straightforward and will not be described here. The comment statements of program DACAL provide a desription of the program.

5.4.2.2 Program CONVWAL. Program CONVWAL converts a selected portion of the NASA digital data tape to a Cyber digital tape. As in previous programs, subroutine GBYTES performs the word conversion. The digital data are reformatted. Each Cyber data record is 5001 words long. The first word contains the record time and is a real number. The remaining 5000 words are integers and contain the digital data.

Program CONVWAL is listed in Appendix VII. Preceding the program listing is a Cyber procedure file used to compile and execute program CONVWAL. The line following the /EOR statement contains the five parameters to be input to program CONVWAL. They are: the time to begin conversion, in seconds from launch (must fall on the second); the time to end conversion, in seconds from launch (must fall on the second); the overall digitizing rate of the NASA data tape (samples/second); the launch time, in hours, minutes and seconds of universal time (again the second must be an integer); and the flight number. The program is designed to provide one second of data per Cyber record for a 5000 samples/second digitizig rate. Combined with the fact that the record must begin on the second, processing of the converted and reformatted digital data tape is simplified.

If only a small portion of the NASA digital tape is to be converted (less than three minutes for a 5000 samples/second digitizing rate or less than 30 seconds for a 25,000 samples/second rate) then the Cyber data file can be stored on disk in a direct access file.

The program is documented with comment statements, so only a synopsis of the program flow will be given here.

After reading the input parameters, the launch time is converted to seconds and the start and end times are converted to seconds of universal time. Then the inut parameters are output for diagnostic purposes.

The tape is positioned at the start of data by skipping over the header and calibration records. Subroutine CALHED performs this function. The record which contains the time to start data conversion is located by successive calls to subroutine RECTIM. This subroutine returns either the time encoded on the current record or the time encoded on the next record depending on the value of logical variable STAY. If a parity error is encountered while reading a record, that record is skipped.

When the record containing the time to start conversion is reached the index of the data point which corresponds to the start time is computed. For the data on the converted tape to be consistent, the channel with which to begin conversion must be the first. The index at which to begin processing is adjusted to correspond to channel one. The number of data words left in the record is computed and then output along with the starting index for diagnostic purposes.

Because the 2008 16-bit IBM words are stored in nearly 536 60-bit Cyber words the starting index must be converted to a bit location in a Cyber word. After this is accomplished the actual data word conversion is performed by calling subroutine TPGET to place the 16-bit IBM words in the lower order bits of the 60-bit Cyber words. If a parity error was encountered while reading the IBM tape record the portion of the 5000 word Cyber data array which should have contained these data are padded with the value 9000. Since the digital data values must fall between 0 and 4095 these data points will be flagged as missing in subsequent Cyber processing.

The remaining 5000 words of the Cyber data record are filled in a manner similar to the above method. Then the time of the Cyber data record (corresponding to the first data point ±0.4 msec due to alignment with data channel 1) along with the 5000 data words are output to the Cyber data file.

The start time of the next Cyber data record is computed. Before repeating the entire locate and convert procedure described above, the time encoded on the current IBM data record is output along with the start time of the next Cyber record. If the start time of the next Cyber record is greater than the stop time, then the program terminates.

The process of locating the index in the IBM data record which corresponds to the Cyber data record start time is repeated with each Cyber record to assure that no time slippage results from a NASA data record of incorrect time length. Because record time and array indices are printed out once each second, a digitizing error on the NASA digital data tape can be detected.

5.4.2.3 Programs DAMED and DAAVG. Programs DAMED and DAAVG are used to reduce the digital data values to decibel power levels. Program DAMED finds the median digital data value of a one second interval and then converts this value to a decibel power level. Program DAAVG computes the average digital data value over a one second interval and then converts this value to a decibel power level.

For a signal which either slowly increases or slowly decreases with time the median value is a better approximation to the real value than the average value, especially in the presence of nonsymmetric noise. For the average value to equal the real value, the sum of positive deviations from the real value must equal the sum of negative deviations from the real value. However, for the median value to equal the real value the number of values greater than the real value must equal the number of values less than the real value, a simpler criterion to meet. Additionally, a data value with an abnormally large deviation from the real value will pull the average value farther from the real value, whereas the median value will weigh it like all the other data values.

The analysis of the differential absorption data for the eclipse flights was done using program DAMED. Program DAAVG is included for reference. It is identical to program DAMED except for the replacement of subroutine FINMED by subroutine AVERAG and the replacement of the word median by the word average in the headings.

Program listings and a procedure file used to run these programs are given in Appendix VIII. The procedure file was used to compute the 5 MHz ordinary attenuator levels for flight 18.1020. Direct access file DA1020 contained the converted digital data output from program CONVWAL.

The eleven input parameters follow the /EOR statement. The first line contains the heading to be printed on the output file. The next line contains the number of the tape channel to be processed. The tape channel assignments are listed in Table 5.1.

The next three lines contain the launch time, the time to begin processing and the time to end processing. All are in universal time. The last six lines contain the decibel power level calibrations which were obtained via program DACAL. The first calibration level is the digital value which corresponds to the -50 dB attenuator setting. The second calibration level corresponds to the -40 dB attenuator setting ... and so forth. The last calibration level corresponds to the 0 dB attenuator setting.

Output from program DAMED consists of a listing of the input parameters followed by a listing of the decibel power levels computed for each second. The time is listed in seconds from launch to ease identification. Along with the decibel power level the corresponding median digital value is listed. The difference between the current decibel power level and the previous decibel power level is listed also.

Program processing is direct. After the program flags and variables are initialized the input parameters are read. They are then output for diagnostic purposes. The subroutine which reads the launch, start and stop times, and also converts them to seconds. The start and stop times are converted to seconds from launch to match the time format of the data file generated by program CONVWAL.

The analysis of each one second interval is centered on the second. Therefore, it is necessary to locate the data point which corresponds to the start time minus 0.5 seconds. Once this data point has been located, each consecutive group of 5000 data points is analyzed without rechecking the time. Program CONVWAL will have checked this already.

During the analysis the validity of the data values is checked. If a data value is out of the range of the 12-bit digitizer resolution (0-4095), it is left out of the analysis. (Recall that program CONVWAL inserted an out of range value for bad data records.) If the number of bad data values encountered during one second of processing is nonzero, the number of bad data values is output.

Processing terminates when the end time is reached or when an end-of-file is reached in the data file.

Earlier versions of program DAMED are discussed in GINTHER AND SMITH (1975) and SLEKYS AND MECHTLY (1970).

5.4.2.4 Program DA2NE. Program DA2NE is virtually identical to program FR2NE. Refer to section 5.4.1.3 for the program discussion. Simply replace all FR (Faraday rotation) variables by DA (differential absorption) variables in Figure 5.10 and Table 5.7 to equate Faraday rotation rate processing to differential absorption rate processing.

The function subroutine which computes Faraday rotation rate as a function of electron density, MFR, is replaced by MDA. Function subroutine MDA computes differential absorption rates as a function of electron density. In program DA2NE the Faraday rotation rate is computed in subroutine NPRNT.

Appendix IX lists the main program section of program DA2NE along with subroutines MDA and NPRNT. All other subroutines are listed in Appendix V.

The one input parameter of program DA2NE which differs from program FR2NE is the experimental differential absorption rate. The experimental differential absorption rate is the slope of the smooth curve which is drawn through the relative extraordinary data points. The relative extraordinary power levels are found by subtracting the ordinary power levels from the extraordinary power levels.

Tables 5.8 and 5.9 list the results of the different stages in the processing of the 2 MHz and 5 MHz differential absorption data for flight 18.1020. Figures 5.12 and 5.13 show the smooth curve which is drawn through the relative extraordinary values listed in Tables 5.8 and 5.9. The corrected (smoothed) values of the relative extraordinary data are used in calculating the experimental differential absorption rate.

A differential absorption rate from Table 5.8 or 5.9 is the slope of the smooth curve in either Figure 5.12 or 5.13 computed at a second. Because of the large deviation of these curves from a straight line the

Table 5.8 18.1020 2 MHz differential absorption.

Time From Launch (sec)	Ordinary Power Level (dB)	Extraordinary Power Level (dB)	Relative* Extraordinary Power Level (dB)	Smoothed Relative Extraordinary Power Level (dB)	Differential† Absorption Rate (dB/sec)	Electron Density m^{-3}
50	-20.151	-30.653	-10.502	-10.550	--	--
51	-20.138	-30.705	-10.567	-10.575	-0.050	2.922×10^8
52	-20.151	-30.847	-10.696	-10.650	-0.088	3.907×10^8
53	-20.151	-31.079	-10.928	-10.750	-0.162	5.513×10^8
54	-20.151	-30.898	-10.747	-10.975	-0.262	6.932×10^8
55	-20.151	-31.246	-11.095	-11.275	-0.288	5.965×10^8
56	-20.151	-31.775	-11.624	-11.550	-0.212	3.457×10^8
57	-20.151	-31.685	-11.534	-11.700	-0.038	4.916×10^7
58	-20.151	-31.852	-11.701	-11.625	0.175	unmatchable
59	-20.151	-31.595	-11.444	-11.350	0.412	unmatchable
60	-20.164	-30.950	-10.786	-10.800	0.700	unmatchable
61	-20.151	-30.073	-9.922	-9.950	1.700	unmatchable
62	-20.138	-27.451	-7.313	-7.400	3.088	unmatchable
63	-20.124	-23.905	-3.781	-3.775	--	--

*$(Ex\text{-}Ord)_{relative} = (Ex\text{-}Ord) - Ord$

† DA rate = $\frac{y_3 - y_1}{2}$ where y_1, y_2 and y_3 are three consecutive smoothed relative extraordinary level points, spaced one second apart.

Table 5.9 18.1020 5 MHz differential absorption.

Time From Launch (sec)	Ordinary Power Level (dB)	Extraordinary Power Level (dB)	Relative* Extraordinary Power Level (dB)	Smoothed Relative Extraordinary Power Level (dB)	Differential† Absorption Rate (dB/sec)	Electron Density m^{-3}
50	-19.705	-30.092	-10.387	-10.350	--	--
51	-19.705	-29.903	-10.198	-10.350	0.000	7.638×10^5
52	-19.721	-30.078	-10.357	-10.350	0.000	6.168×10^5
53	-19.721	-30.105	-10.384	-10.350	-0.025	1.264×10^8
54	-19.705	-30.092	-10.387	-10.400	-0.125	5.282×10^8
55	-19.721	-30.293	-10.575	-10.600	-0.188	6.728×10^8
56	-19.721	-30.576	-10.855	-10.775	-0.112	unmatchable
57	-19.721	-30.455	-10.734	-10.825	0.038	1.019×10^8
58	-19.721	-30.468	-10.747	-10.700	0.175	4.190×10^8
59	-19.737	-30.240	-10.503	-10.475	0.250	5.447×10^8
60	-19.737	-30.038	-10.301	-10.200	0.338	6.879×10^8
61	-19.737	-29.527	-9.790	-9.800	0.475	9.070×10^8
62	-19.737	-28.975	-9.238	-9.250	0.725	1.313×10^9
63	-19.689	-27.979	-8.290	-8.350	1.175	2.082×10^9
64	-19.689	-26.458	-6.769	-6.900	1.238	2.230×10^9
65	-19.769	-24.573	-4.804	-5.875	1.900	3.507×10^9
66	-19.689	-22.783	-3.094	-3.100	--	--

*$(\text{Ex-Ord})_{\text{relative}} = (\text{Ex-Ord}) - \text{Ord}$

†DA Rate = $\frac{y_3 - y_1}{2}$ where y_1, y_2 and y_3 are three consecutive smoothed relative extraordinary level points, spaced one second apart.

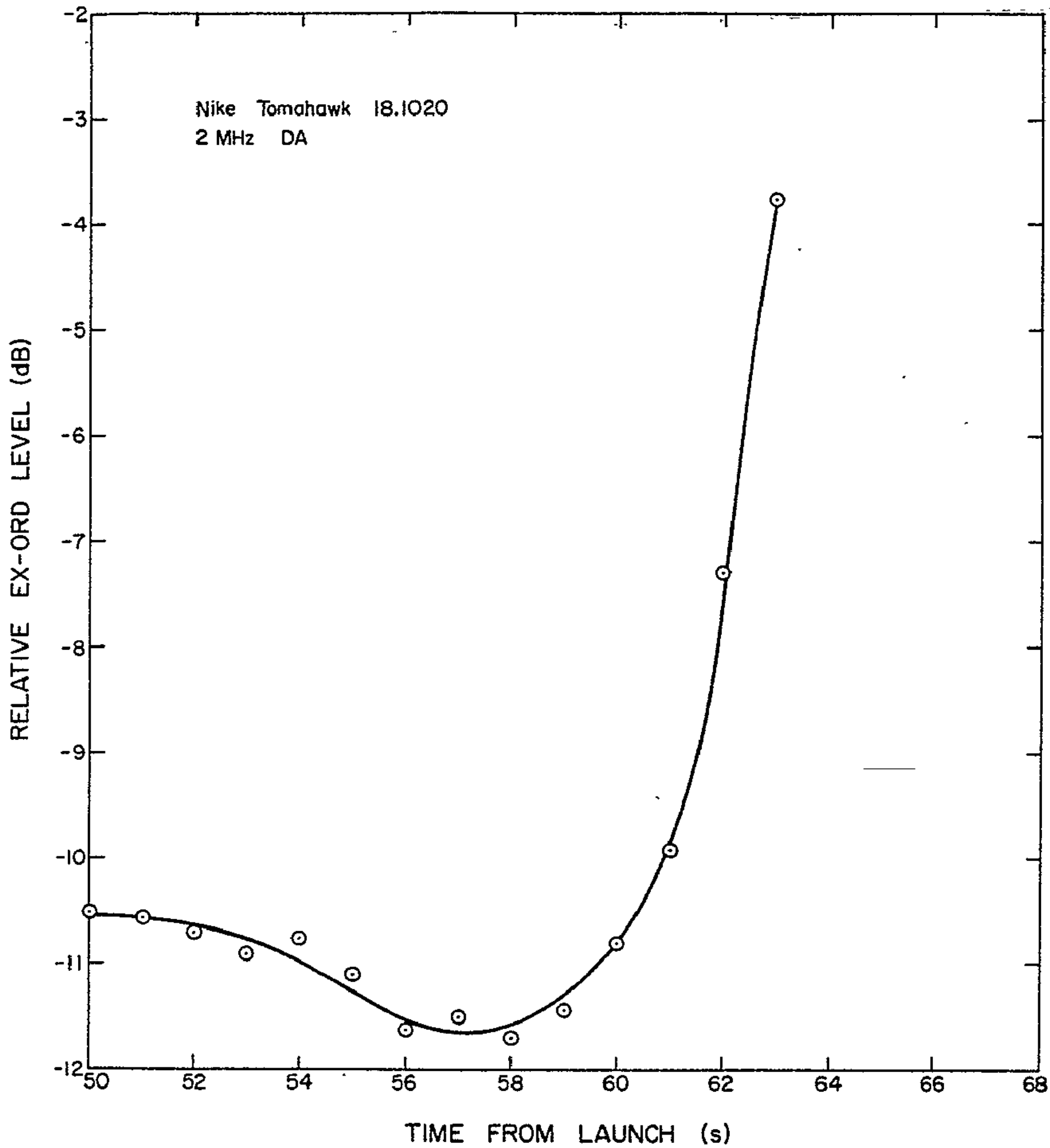


Figure 5.12 Plot of relative Ex-Ord levels from the 2 MHz differential absorption experiment for flight 18.1020. The derivative of the smooth curve which has been drawn through the points is the differential absorption rate.

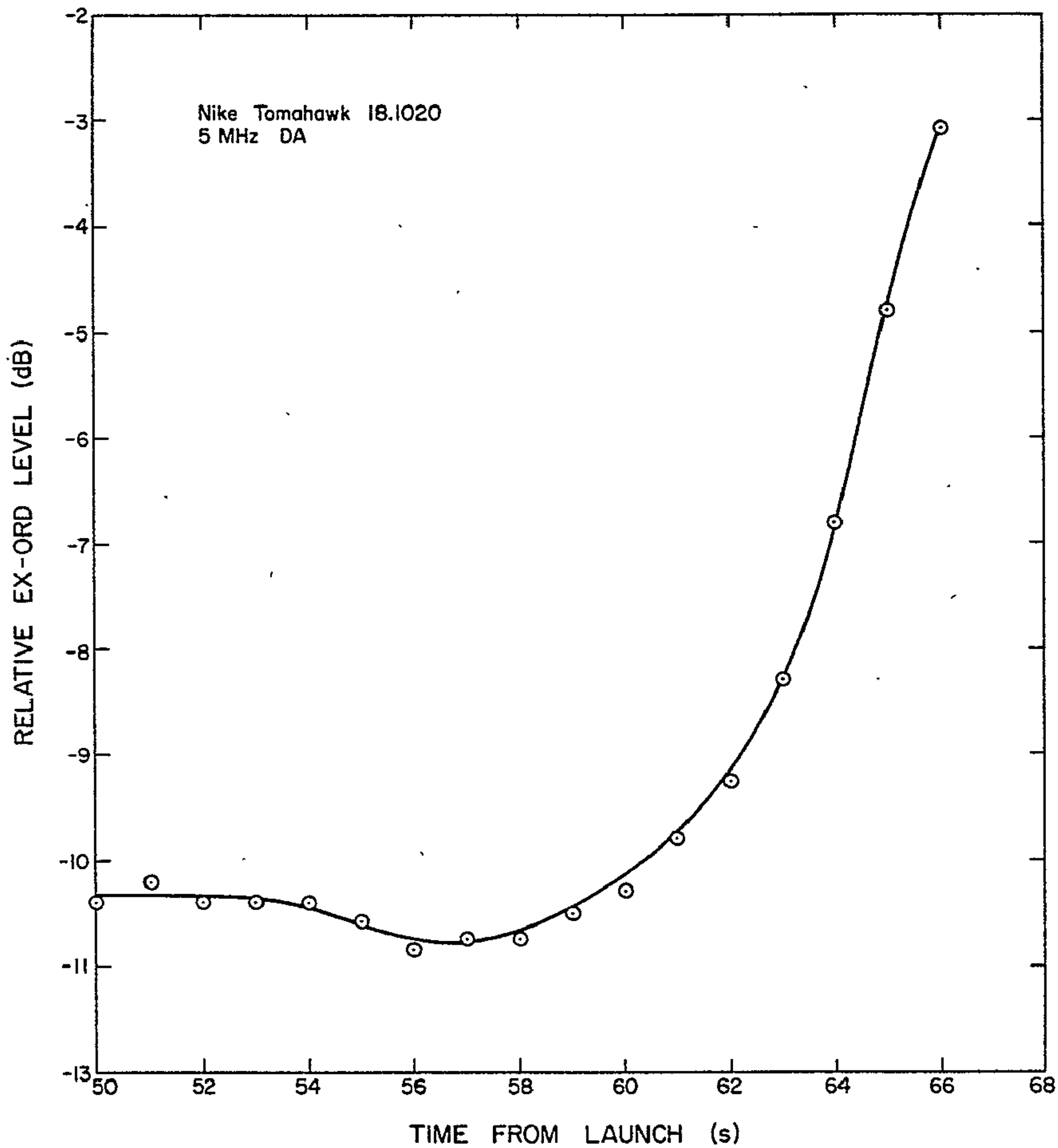


Figure 5.13 Plot of relative Ex-Ord levels from the 5 MHz differential absorption experiment for flight 18.1020. The derivative of the smooth curve which has been drawn through the points is the differential absorption rate.

approximation to the slope by differencing two consecutive points is not accurate. A more accurate approximation is the slope of the parabola which has been fit through three consecutive points. In other words: A parabola is fit to three consecutive points, which are spaced one second apart; the slope of the parabola at the center point is the differential absorption rate for that time.

For three points (t_1, y_1), (t_2, y_2) and (t_3, y_3) with a spacing of one second (i.e., $t_1 + 1 = t_2 = t_3 - 1$) the slope of the parabolic curve fit through these points is given by

$$\left.\frac{dy}{dt}\right|_{t = t_2} \doteq \frac{y_3 - y_1}{2} \tag{5.4}$$

This formula was used to compute the experimental differential absorption rates for the eclipse flights.

Figure 5.14 is a plot of the electron densities which were matched to the experimental differential absorption rates by program DA2NE for flight 18.1020. These electron densities are listed in Tables 5.8 and 5.9.

5.5 Electron Density Profile

To determine the final electron density profile the probe current profile must be calibrated by the propagation experiment; the gaps in the profile must be filled; and finally, the profile must be plotted. The three computer programs used to perform these tasks are: PC2ED, FDINTER and EDPLOT. Each will be described in this section. Program listings may be found in the appendices.

5.5.1 Program PC2ED. Program PC2ED performs the simple task of multiplying the probe current profile by a constant N/I factor to convert probe current to electron density. The program along with the procedure file used to run the program are given in Appendix X.

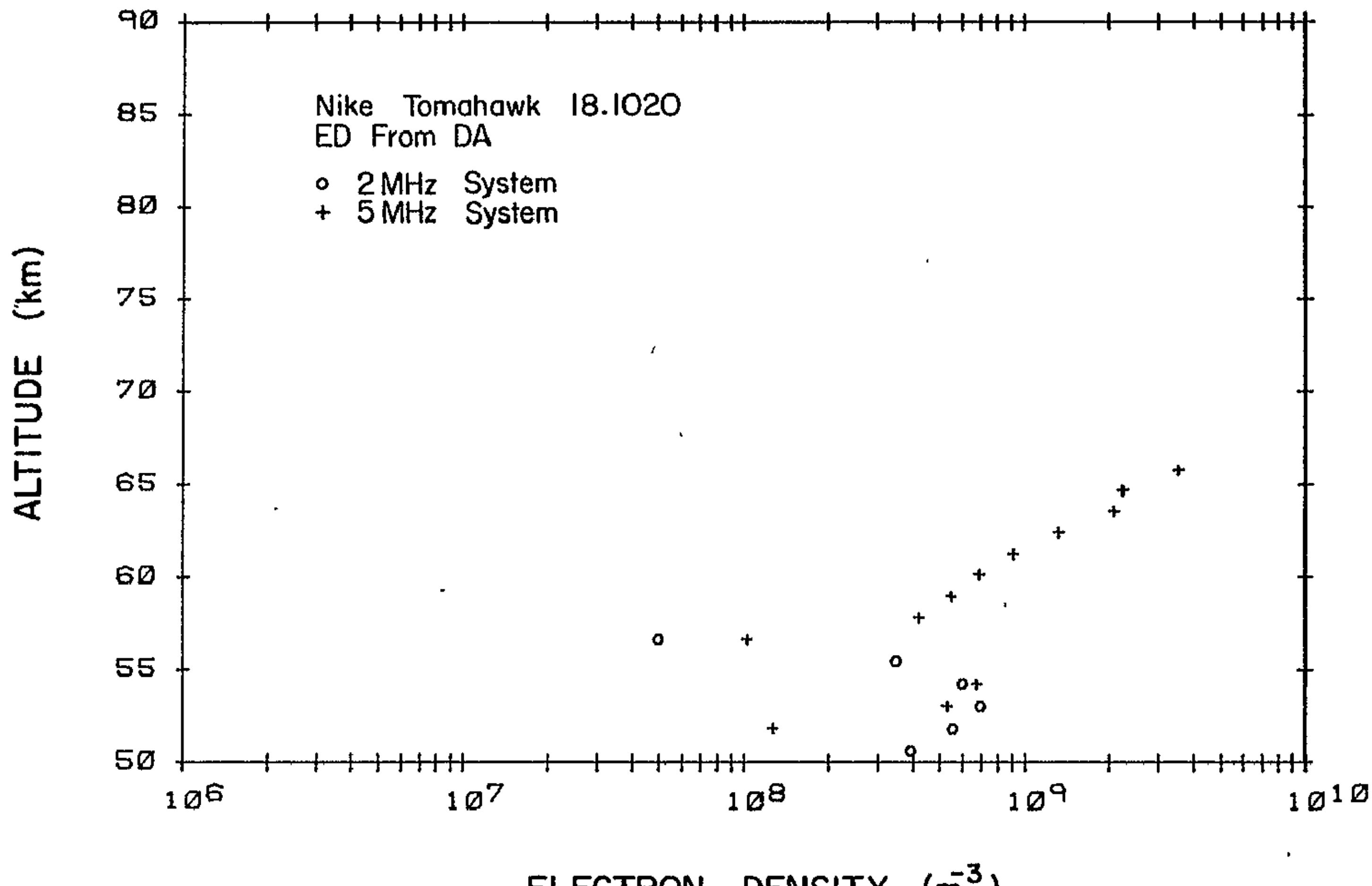


Figure 5.14 Plot of electron densities derived from differential absorption rates for flight 18.1020. The circles represent the 2 MHz experimental results while the crosses represent the 5 MHz experimental results.

Because the propagation data are very limited in altitude a constant N/I value is used to calibrate the eclipse probe profiles. Usually the probe profile is calibrated for each individual second in the D region and a constant N/I value is not applied until within the E region. (Recall that there is some uncertainty associated with the Langmuir probe theory in the D-region).

The program has the option of using two calibration factors. The three input parameters are: the first N/I value; the time from launch, in seconds, up to which the first N/I value will be used; and the second N/I value. These three parameters follow the /EOR procedure file statement in Appendix X.

The first N/I value was used up until the drop in probe current at separation. It was determined by placing the electron density plot from the differential absorption experiment over the probe current profile and "fitting" by eye. The ratio of electron density to probe current at two coincident X-axis tick marks was used as the first N/I value.

The second calibration factor was determined by computing the ratio of probe current directly before separation to probe current immediately following separation and multiplying the first calibration value by this ratio. These two probe current values along with the approximate time at which separation occurred can be found by examining a listing of the probe current values.

A plot of the calibrated probe profile for flight 18.1020 is shown in Figure 5.15. Both ascent and descent are plotted. Note the glitch which still remains at separation. Program PC2ED is very short, simple and well commented so it will not be described here.

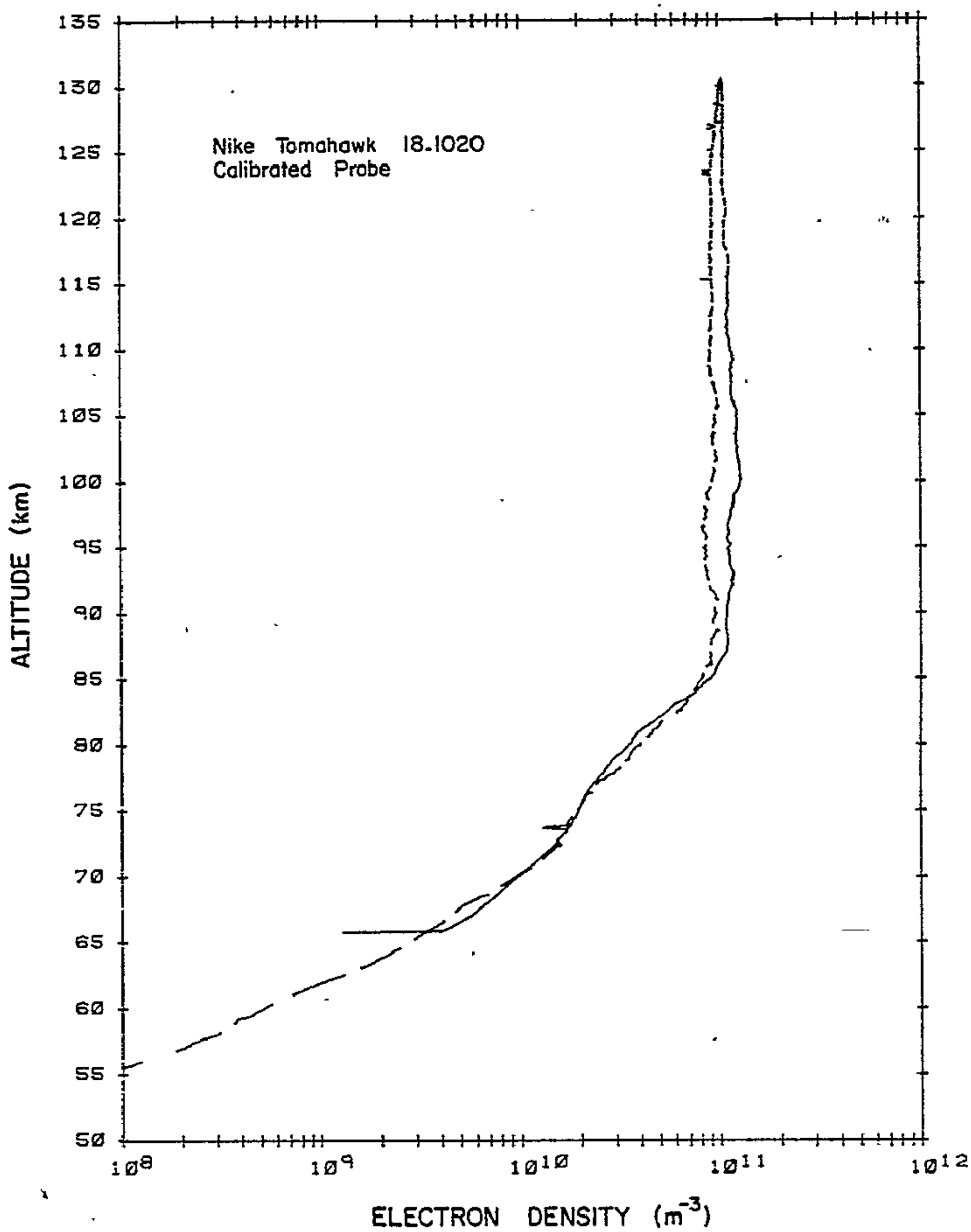


Figure 5.15 Calibrated probe profile for flight 18.1020. The electron densities derived from the differential absorption experiment were used to determine the calibration factor.

5.5.2 Program EDINTER. The gaps in the calibrated probe current profile are filled by program EDINTER. A linear interpolation is applied using the two electron density values which bracket a gap as bases for the interpolation. Because the electron density in the ionosphere changes slowly with altitude and the sweeps cover a short altitude range, a linear interpolation is adequate. The one altitude range which is an exception is the lower E-region between 90 and 105 km. This is where sporadic E (a large gradient in electron density) can occur. In order to examine continuously the electron density of this region, the sweeps are not initiated until 120 km on ascent.

Appendix XI contains the listing of program EDINTER. The first thirteen statements are the Cyber procedure file used to run program EDINTER for flight 18.1020. There are no input parameters for this program. However, the calibrated probe current profile should be "cleaned up" first. The major glitches in the profile can be removed by replacing their electron density values with the flag value of 2. Program EDINTER interpolates over 2's.

From a plot of the calibrated probe current profile and a listing of the data file the major glitches can be manually edited.

Figure 5.16 is a plot of the electron density profile of flight 18.1020. Note that the glitches of Figure 5.15 have been removed. Also note that it is difficult to discern the ascent portion of the profile from the descent portion. For this reason the uninterpolated descent profile (with the glitches removed, but not filled) was used in the final electron density plot of each flight. Since program EDINTER is short and straightforward it will not be described here.

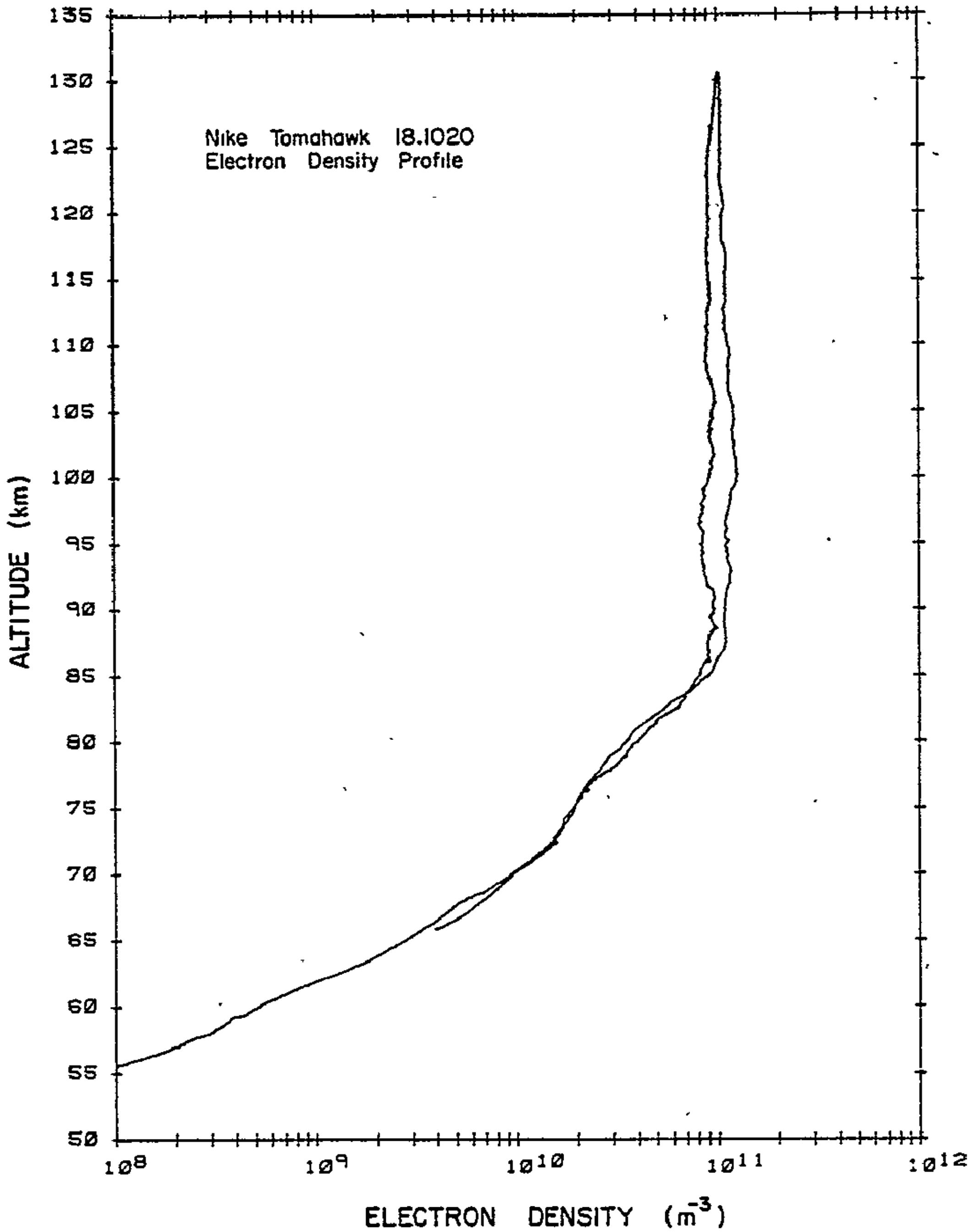


Figure 5.16 Final electron density profile for flight 18.1020. The glitches and gaps of Figure 5.15 have been removed and filled.

5.5.3 Program EDPLOT. Program EDPLOT is a general purpose plotting program. In addition to electron density profiles, probe current profiles and individual electron density values can be plotted. The option of plotting more than one profile per axis is also provided, but only for Zeta (paper) plots.

Appendix XII lists program EDPLOT along with four sample execution sequences. Program EDPLOT is interactive and must be executed in the time-sharing mode. The program is generally used to produce hard copy (Zeta) plots. Each of the four sample execution sequences produces a Zeta plot.

The third execution sequence is an example of producing a final electron density plot. A final electron density plot has a standard size and scale. The size is 10 inches by 15 inches with a horizontal scale of 10^1 to 10^6 cm^{-3} (10^7 to 10^{12} m^{-3}) and a vertical scale of 50 to 200 km. These standards are used to ease comparisons of electron density profiles. The first electron density profiles (prior to 1979) were not created using the Cyber computer and hence are not conveniently stored on the computer system. This makes it necessary to duplicate the standard size which was originally set for final electron density plots.

All electron density data files after 1979 are stored on the computer system. Program EDPLOT permits the plotting of profiles from two such files on the same axis. The fourth execution sequence of Appendix XII is an example of superimposing the electron density profiles of flights 18.1020 and 18.1021.

Although the size of each plot is fixed for each plotting device (set by subroutine DEVICE) the horizontal and vertical scales are selectable. For data files which cover an entire flight separate plotting of ascent and descent is available.

The bulk of program EDPLOT consists of asking the user for plotting parameters and checking the validity of the values entered for these parameters. These functions are quite straightforward. However, the actual plotting operation is somewhat complicated. Figure 5.17 is a flowchart of the plotting section of program EDPLOT. Table 5.10 defines the variables and subroutines found in Figure 5.17. Program SWEEP contains a similar plotting section.

The Graphics Compatibility System (GCS) plotting routines are used in this program. The routines are device independent. To change plotting devices it is required only to "GRAB" a different device driver (e.g., TEKT for a Tektronics terminal or ZETA for a Zeta drum plotter).

The first page of Figure 5.17 and the first column of page two provide the steps to define the plotting area and draw the axes. For a plot done on the Zeta drum plotter using 8 1/2" x 11" fanfold sheets, the left and bottom margins are reduced by 1/4" to position the pen farther from the perforation.

The reason for the complicated plotting procedure is the need to skip over values of 1 and 2 in the data file. Although the pen is lifted, a valid plot point must be substituted or a "pen out of bounds" error might result from attempting to move to a point with a horizontal position of 1 or 2. Variable PRENDX holds the index in array PC of the last value which was neither 1 nor 2.

In the case of an isolated plot point (such as the individual points of a derived electron density plot which results from the propagation experiment) the steps on page four column two of the flowchart are executed. Either a circle or a cross is plotted for the point: A circle for a point in input file DATA1 and a cross for a point in input file DATA2.

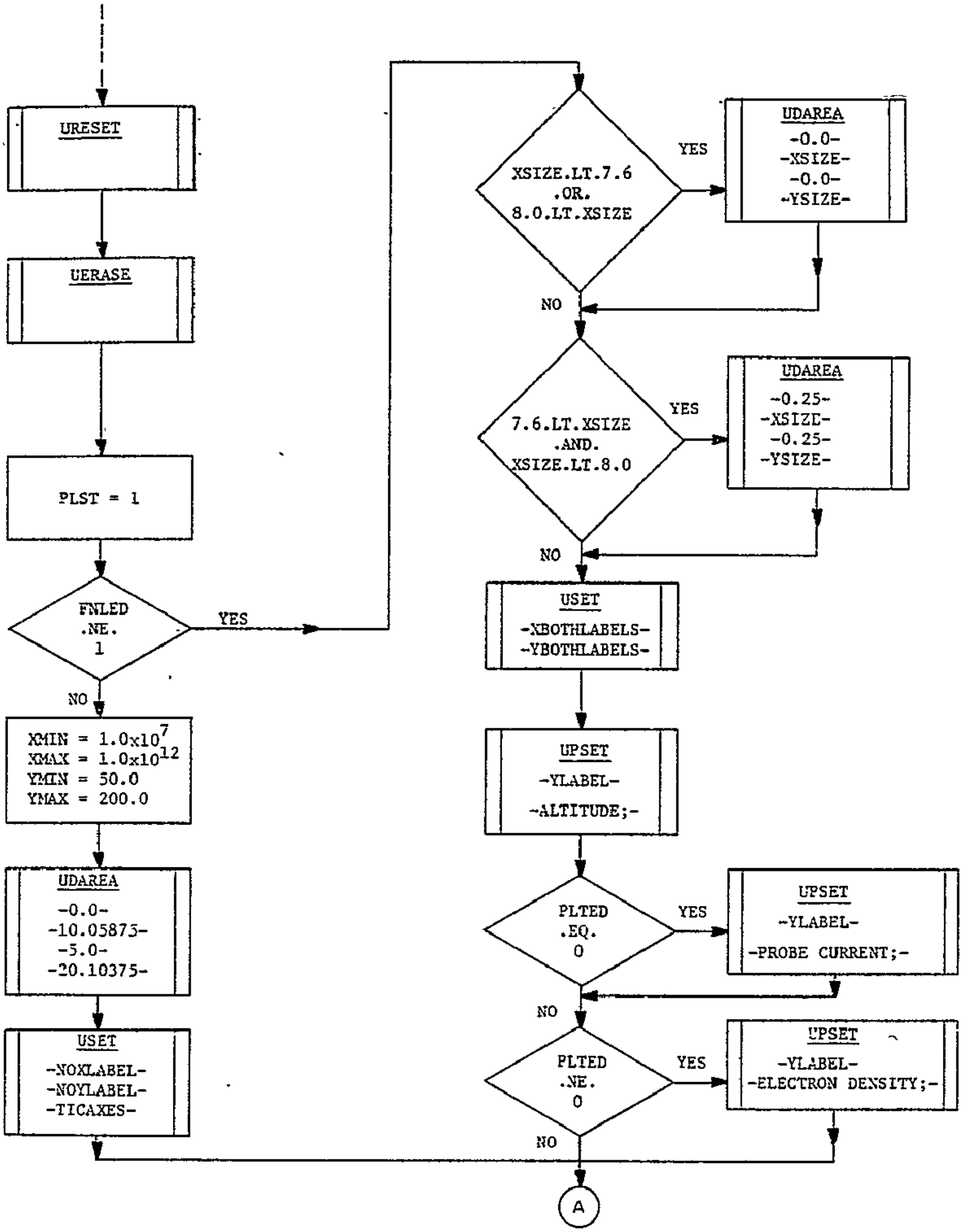


Figure 5.17 Flowchart of the probe current/electron density plotting section of program EDPLOT. Table 5.10 defines the variables and subroutines found in this figure.

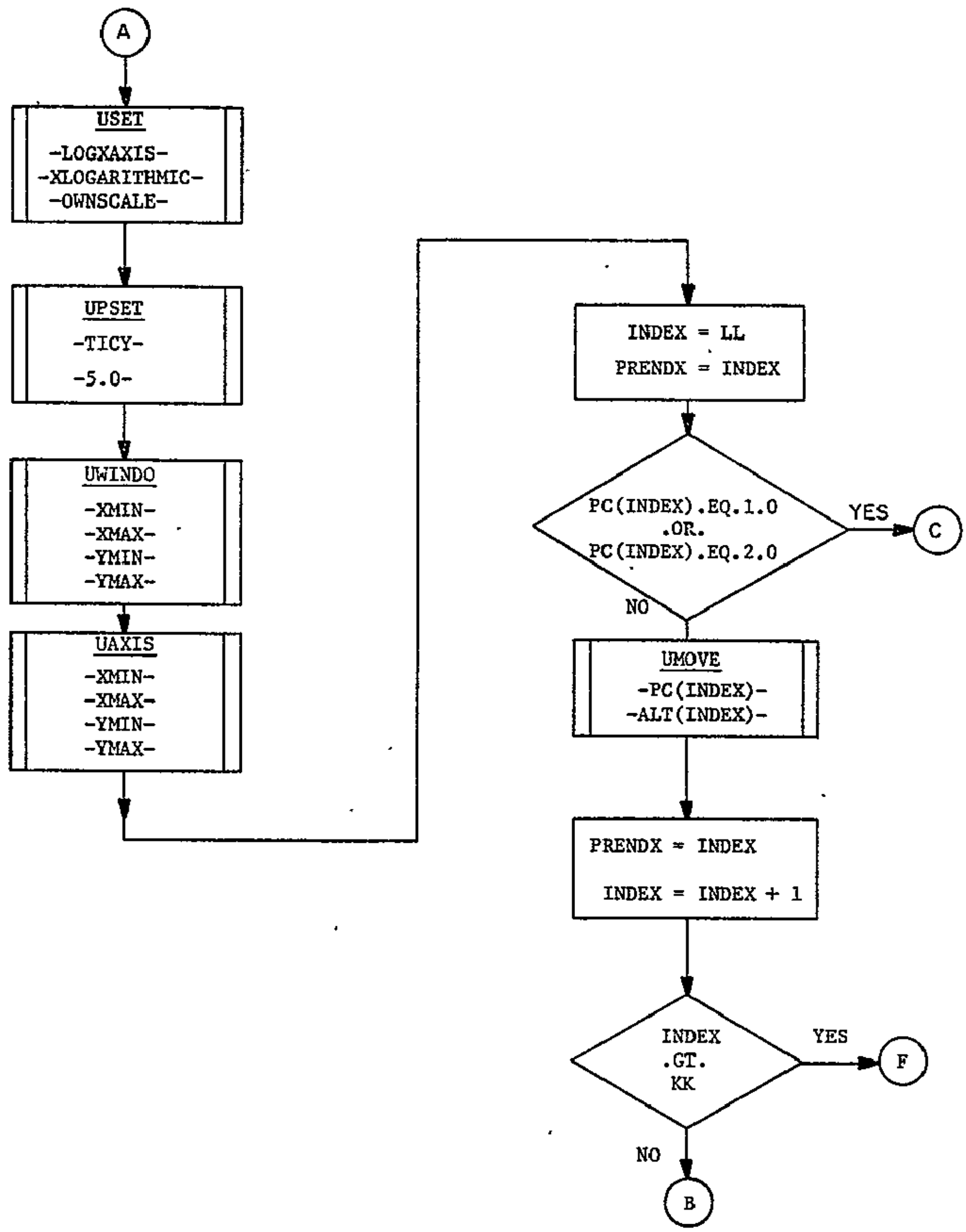


Figure 5.17 (Continued).

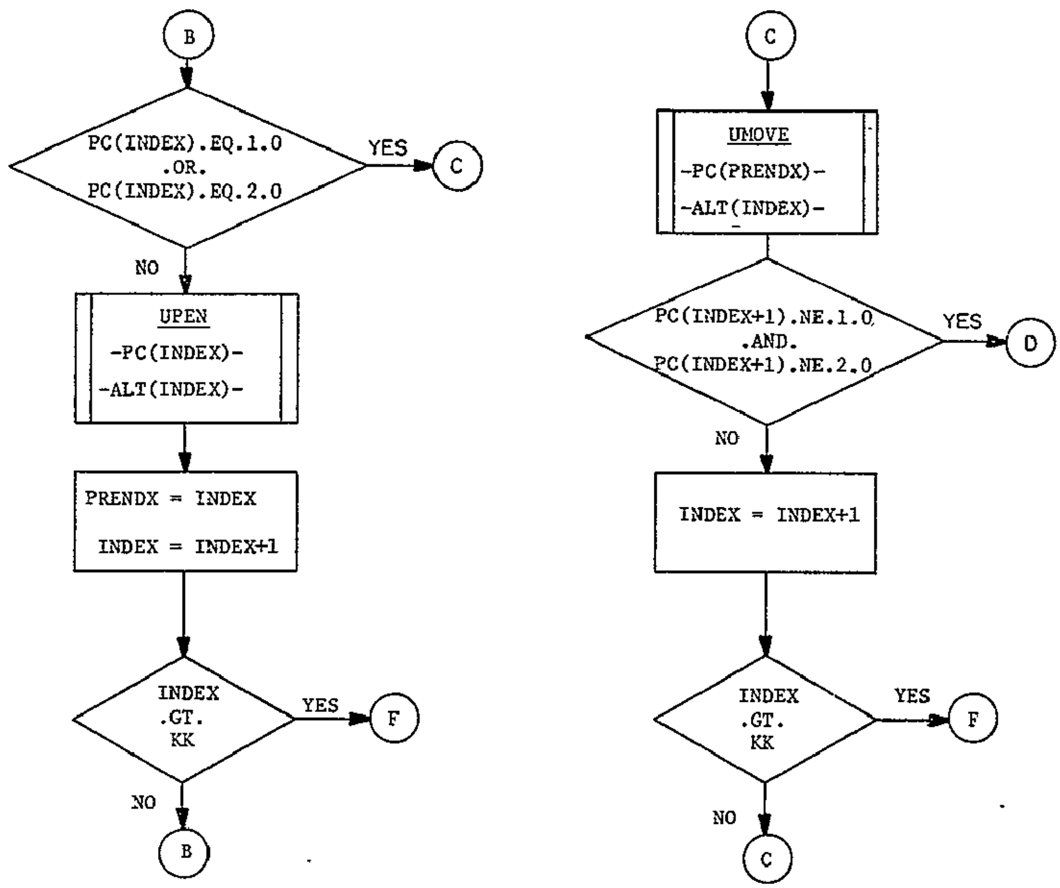


Figure 5.17 (Continued).

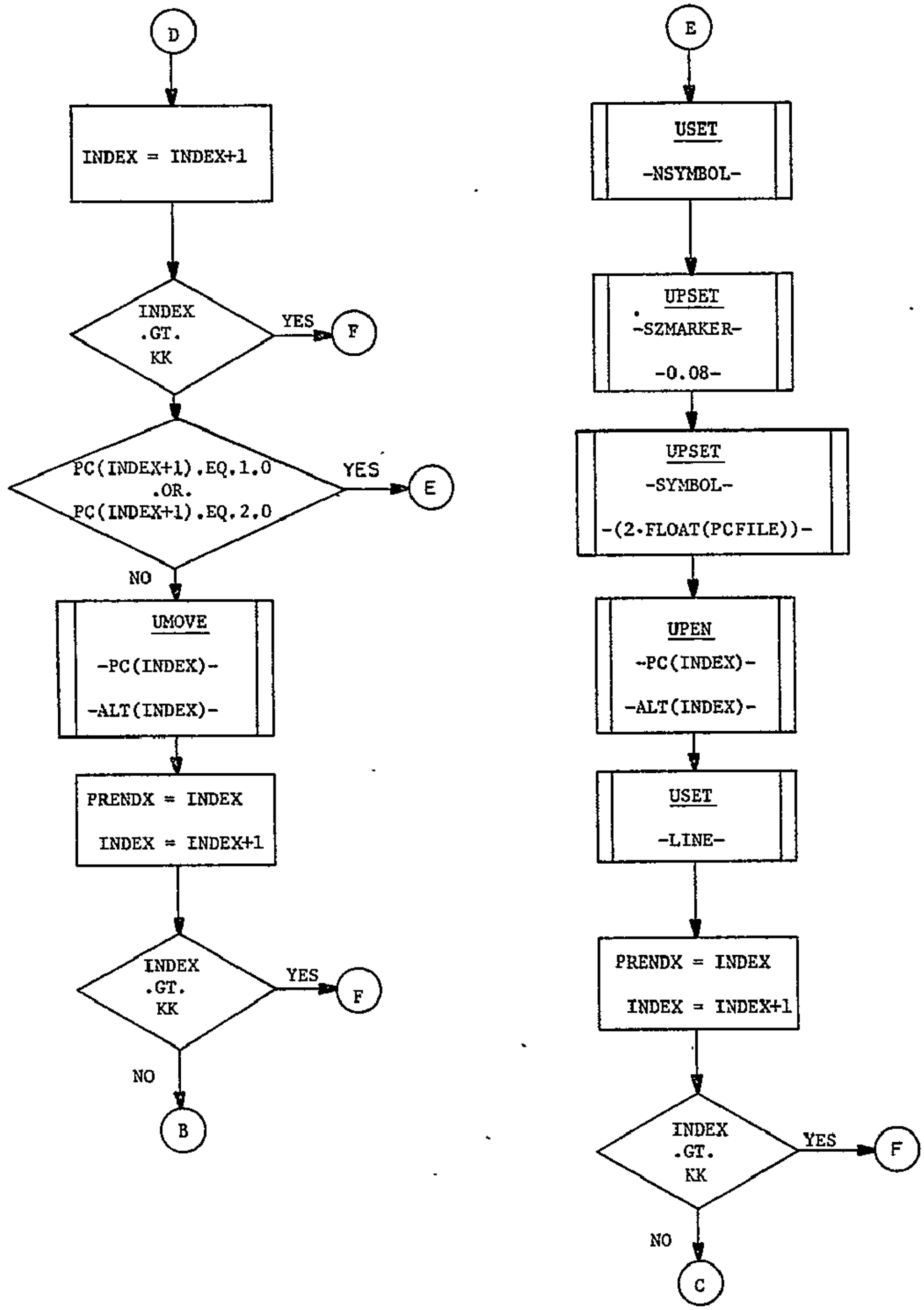


Figure 5.17 (Continued).

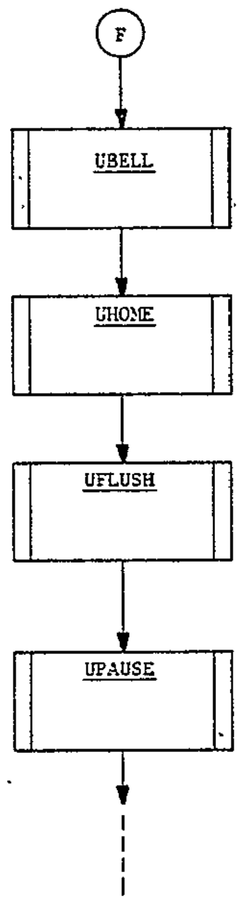


Figure 5.17 (Continued).

Table 5.10 Variables and subroutines in program EDPLOT.

UREBET - GCS subroutine: This subroutine resets GCS variables to their default conditions.

UERASE - GCS subroutine: This subroutine erases the plotting screen. For paper plots it advances to the next sheet.

PLST - Flag: 0 indicates that no plotting has been done yet; 1 indicates that plotting has been done.

FNLED - Flag: 0 indicates regular plot; 1 indicates final electron density plot.

XMIN - Minimum X-axis value.

XMAX - Maximum X-axis value.

YMIN - Minimum Y-axis value.

YMAX - Maximum Y-axis value.

UDAREA - GCS subroutine: This subroutine defines the maximum plotting area. In this program the parameters are in inches.

USET - GCS subroutine: This subroutine sets GCS plotting parameters.

XSIZE - Horizontal size of plotting surface, in inches. Its value is dependent upon the plotting device being used and is set by subroutine DEVICE.

YSIZE - Vertical size of plotting surface, in inches. Its value is dependent upon the plotting device being used and is set by subroutine DEVICE.

UPSET - GCS subroutine: This subroutine assigns values (numerical or character) to GCS plotting variables.

PLTED - Flag: 0 indicates plot of probe current; 1 indicates plot of electron density.

UWINDO - GCS subroutine: This subroutine sets the minimum and maximum horizontal and vertical values for plotting in virtual space (i.e., XMIN is equated to the minimum horizontal position; XMAX is equated to the maximum horizontal position; etc. Minimum and maximum positions were set by UDAREA).

UAXIS - GCS subroutine: This subroutine draws axes, with labels and tic marks if requested. The four call parameters define the minimum and maximum values on the X- and Y-axes.

INDEX - Index in arrays PC and ALT of the current point to be plotted.

LL - Index in arrays PC and ALT of the first point to be plotted.

PRENDX - Index in arrays PC and ALT of the previous plot point which was unequal to 1 or 2.

PC - Array of probe current or electron density values.

UMOVE - GCS subroutine: This subroutine moves the plotting pen to the position specified by the two call parameters.

ALT - Array of altitude values.

KK - Index in arrays PC and ALT of the final point to be plotted.

UPEN - GCS subroutine: This subroutine draws a line from the current pen position to the position specified by the call parameters.

UBELL - GCS subroutine: This subroutine sounds the bell on the plotting device.

UHOME - GCS subroutine: This subroutine positions the pen at the home position.

UFLUSH - GCS subroutine: This subroutine makes sure that all GCS commands have been sent to the plotting device.

UPAUSE - GCS subroutine: For plots on a terminal this subroutine suspends program execution until a key is pressed. It is disabled for Zeta plots.

All of the probe current profiles, derived electron density plots and electron density profiles for the 1979 eclipse flights which appear in this thesis were plotted using program EDPLOT.

5.6 Energetic Particle Analysis

There is one general purpose program used to extract the energetic particle count rates, program EEDPROC. In addition to the program's main task of deriving detector count rates it can also calculate the rocket's azimuth angle. The rocket's azimuth angle is used to determine detector orientation. The changing amplitude of the magnetometer signal is the basis for the azimuth angle calculation.

5.6.1 Program EEDPROC. Program EEDPROC is listed in Appendix XIII. The JCL used to run the program on the IBM for flight 18.1021 is included. There are fourteen input parameters for the program: eleven of which are input via namelist. The PARMS namelist contains the number of header records, NUMHED; the number of calibration records, NUMCAL; and a diagnostic level parameter, ICHECK.

The CHANS namelist contains the data tape channel assignments for the five EED channels which the program can analyze. The processing of each EED channel is not the same, so assignment is important.

The TIMES namelist contains the times to begin and end processing, FSTSEC and LSTSEC, respectively. These times are entered in seconds from launch.

The launch time is the next parameter input. It has units of hours, minutes and seconds of universal time.

The next input line holds the data identification information. Up to eighty characters can be entered.

The last two input parameters are the digital numbers assigned to a level 15 count, ICMAX, and to a level 0 count, ICMIN.

Output from program EEDPROC consists of detector count rates in counts per 0.2 sec (due to the 25 kHz digitizing rate) in both printed and punched forms; a printed listing of the program input parameters; and various diagnostics. The level of the diagnostics, from none to extensive, is set by input parameter ICHECK. ICHECK can range from 0 (no diagnostics) to 3 (extensive diagnostics).

In analyzing the eclipse data, the azimuth angle feature of the program was not used. It was suppressed by changing the program statement which called the ZANGLE subroutine into a comment statement.

Program EEDPROC was run twice for each flight. First to assign a digital value to each count level and then to compute the actual count rates.

The digital value which is assigned to each count level is determined from a dump of the data tape over a selected time interval. When ICHECK is set equal to 3 the program will dump the data from each tape record over the interval defined by FSTSEC and LSTSEC.

The last stage of the energetic particle experiment aboard the payload is a 4-bit binary counter. Because the telemetry system is analog, the digital value of this counter is converted to an analog voltage for transmission to the ground-based telemetry station. As the counter increments, the analog voltage increases in steps. Because it is a 4-bit counter there are 16 steps.

The process of assigning a digital value to each step is accomplished by scanning the tape dump and noting at which digital values the steps occur. A determination of the noise within the system is also made. Due to noise the digital value of each level drifts slightly, usually by not more than ±2. If the noise is ever so great as to make identification of individual levels impossible, then there is no use continuing the analysis.

Because the counter resets to zero after fifteen, the analog output will look like individual staircases of 16 steps with a variable step length. The chart record can be examined for a time interval when the stairsteps are prominent on each channel. This time interval is then used for the calibration run of program EEDPROC.

The level 15 and level 0 digital values are used as input for the final run of program EEDPROC. A diagnostic level of 1 is usually used for the final run. Again the chart record can be examined to find the time interval of useful data.

The computation of count rates is done between statement labels 310 and 525 in program EEDPROC. The EED_I arrays contain 1000 data values from each of the five channels. The number of counts per 1000 data samples is performed by first counting the number of transitions from step 15 to step 0 (each 15 $\rightarrow$ 0 transition indicates a count of 16); multiplying that number of transitions by 16 to convert to total counts; and then subtracting from these counts the initial counterstep ($EED_I(1)$) and adding to those counts the final counter step ($EED_I(1000)$). The criterion for defining a step 15 to step 0 transition varies with channel and has been determined empirically.

The number of counts per second are plotted (with a 25 kHz digitizing rate each consecutive group of five output count rates must be summed to convert to counts per second). Figures 6.14, 6.15 and 6.16 are plots of the counts per second for flights 18.1020, 18.1021 and 18.1022, respectively.

6. DATA SUMMARY AND CONCLUSIONS

The final experimental results are presented in section 6.1 for the probe experiment; section 6.2 for the propagation experiment and section 6.4 for the particle experiment. The calibrated probe profiles, with the sweep gaps filled, are presented in section 6.3. A brief discussion of the technique used to match the experimental Faraday rotation rates and the experimental differential absorption rates to electron densities is given in section 6.2.

The experimental results are discussed in section 6.5. Comparisons are made to previous electron density measurements.

Section 6.6 suggests a few research topics which arise from the unique conditions of these flights.

6.1 Probe Profiles

The probe profiles for flights 18.1020, 18.1021 and 18.1022 are plotted in Figures 6.1, 6.2 and 6.3, respectively. These plots include both ascent and descent data, with the sweeps removed.

On each profile there is a sharp decrease in probe current at approximately 75 km on descent. This decrease, by a factor of 1.5, occurs when the payload separates from the Tomahawk motor. The separation of the Tomahawk motor changes the area ratio of the probe to the rocket body, which might affect the probe current through a change in vehicle potential (SMITH, 1969).

Two probe experiment characteristics which can be seen in these profiles are boom deployment at 65 km and the initiation of the sweep circuitry at 115 km.

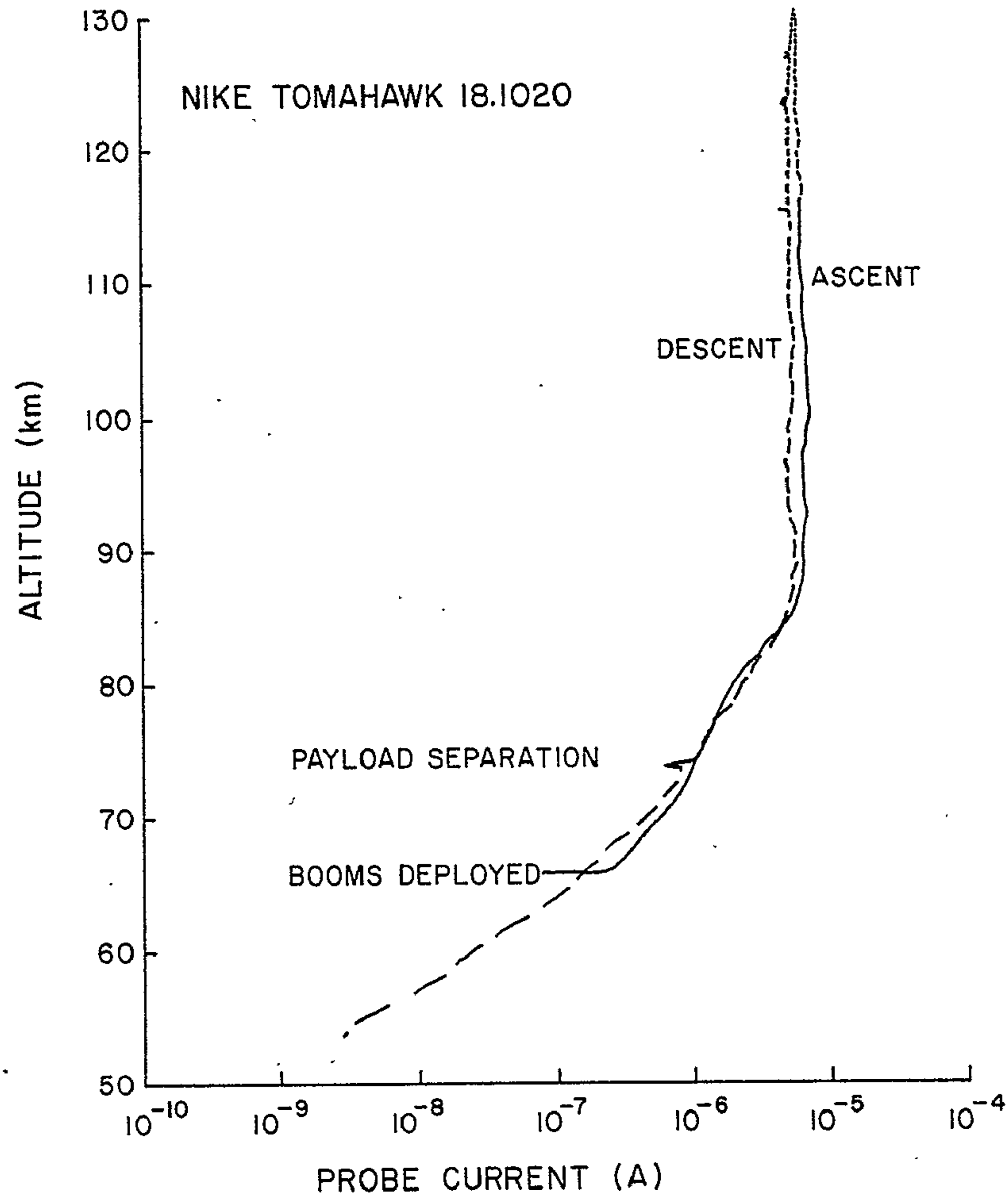


Figure 6.1 Profile of probe current from Nike Tomahawk 18.1020, launched at 1052 LST (1652 UT) on 24 February 1979.

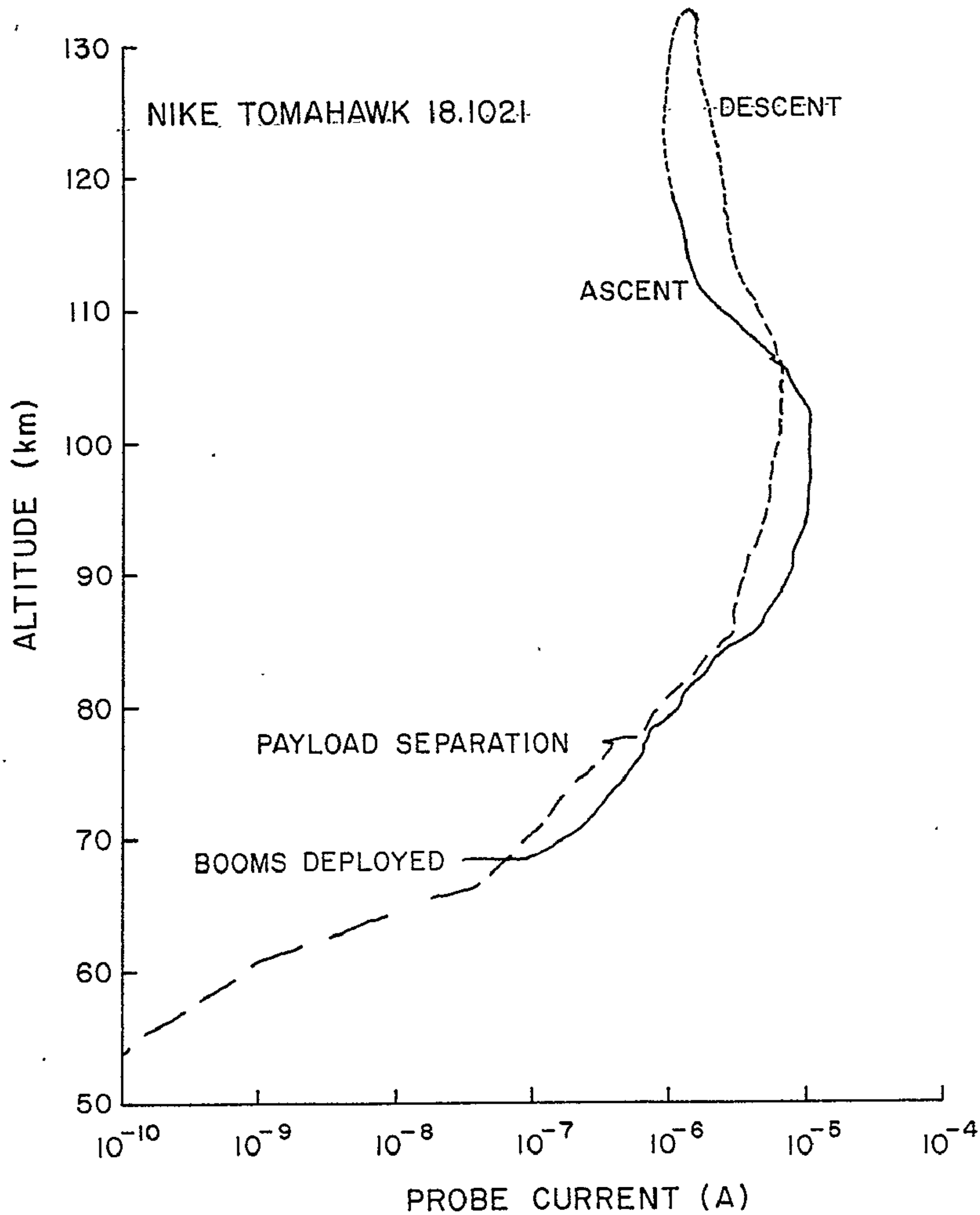


Figure 6.2 Profile of probe current from Nike Tomahawk 18.1021, launched at 1052 LST (1652 UT) on 26 February 1979, during the eclipse.

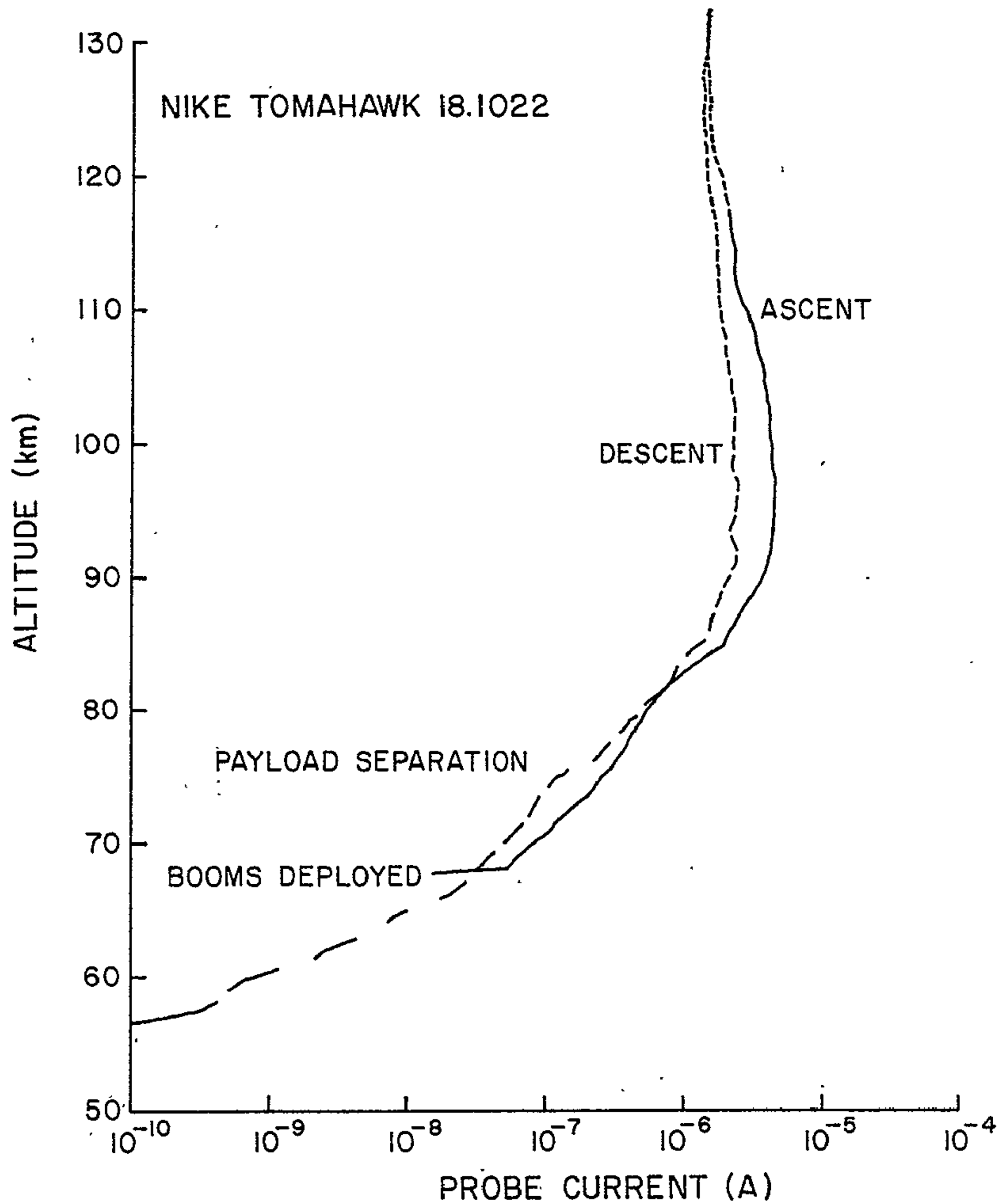


Figure 6.3 Profile of probe current from Nike Tomahawk 18.1022, launched at 1054:10 LST (1654:10 UT) on 26 February 1979, during the eclipse.

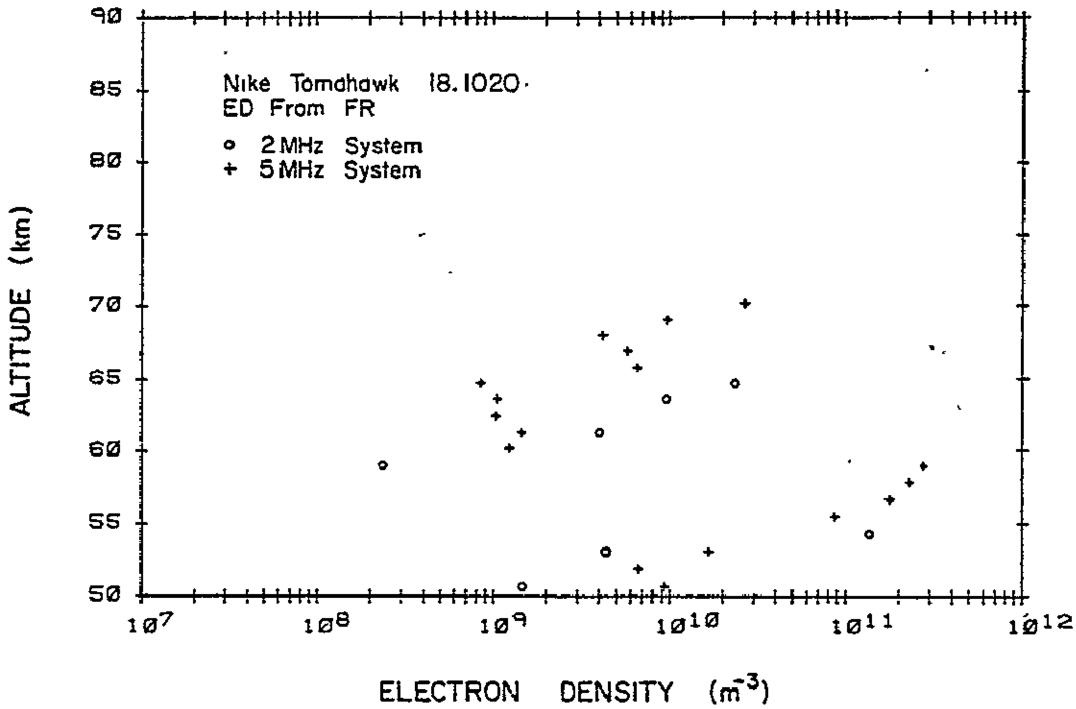
6.2 Propagation Data

6.2.1 Electron density values. Figures 6.4, 6.5 and 6.6 present plots of the electron density values for flights 18.1020, 18.1021 and 18.1022, respectively. Section a of each figure is a plot of the electron density values derived from the Faraday rotation experiment. Section b of each figure is a plot of the electron density values derived from the differential absorption experiment. Tables 6.1, 6.2 and 6.3 list the experimental rates and their matched electron densities for each flight. Again, section a is data from the Faraday rotation experiment and section b is data from the differential absorption experiment.

In each of the plots of Figures 6.4 through 6.6 the circles represent the 2 MHz experimental values of electron density and the crosses the 5 MHz values. Data were unobtainable for the 2 MHz Faraday rotation experiment of flight 18.1022 due to a warped digital data tape. Three data points from the 5 MHz Faraday rotation experiment for flight 18.1022 were unavailable due to parity errors encountered while reading the digital data tape.

The reason for the abnormally high values of electron density found near 60 km is unknown. The probe profiles show a uniform decrease in electron density below 65 km. Rocket despin occurs near 55 km on ascent, at 55 seconds after launch. This sudden change in the spin rate affects the Faraday rotation experiment since this experiment is based on a frequency-difference scheme. However, the differential absorption experiment should be uninfluenced by this change in the rocket spin rate. Therefore this phenomenon is not believed to be an artifact of the experimental technique. It may, however, be associated with the interpretation of the Sen-Wyller generalized magnetoionic theory at lower altitudes.

For each flight the differential absorption data indicates a drop in a)



b)

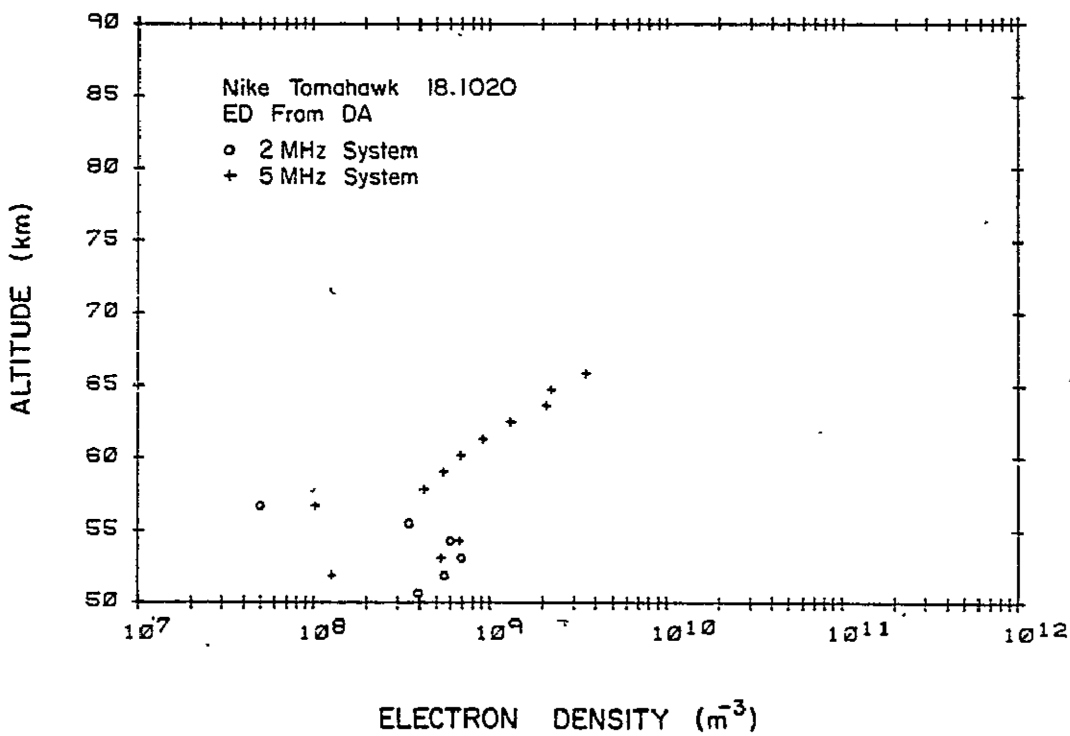
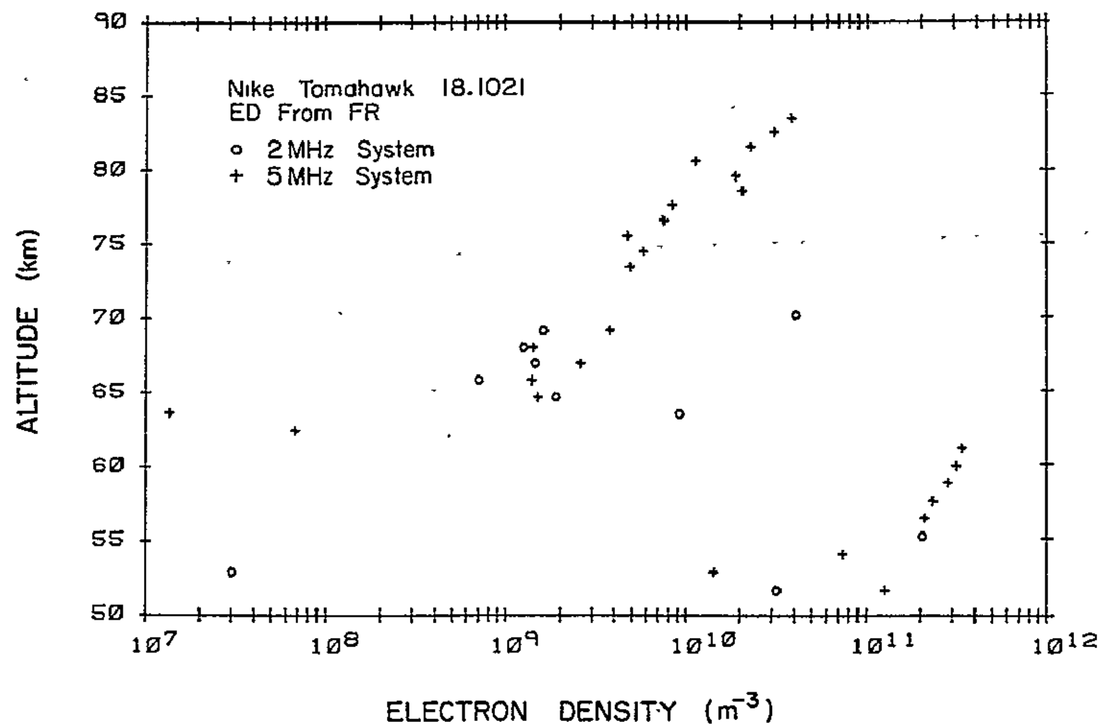


Figure 6.4 Electron density plots from Nike Tomahawk 18.1020, launched at 1052 LST (1652 UT) on 24 February 1979. Electron densities derived from the Faraday rotation experiment are shown in (a). Electron densities derived from the differential absorption experiment are shown in (b). In both (a) and (b) the circles represent 2 MHz measurements and the crosses represent 5 MHz measurements.

a)



b)

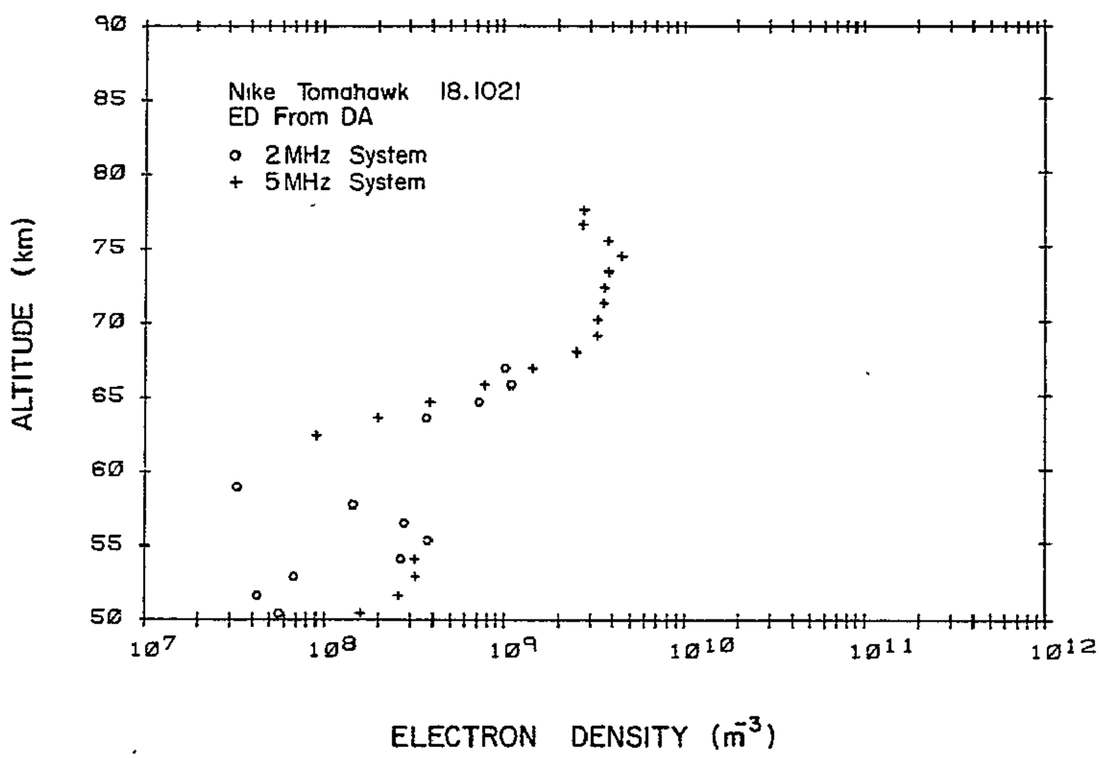


Figure 6.5 Electron density plots from Nike Tomahawk 18.1020, launched at 1052 LST (1652 UT) on 26 February 1979, during the eclipse. Electron densities derived from the Faraday rotation experiment are shown in (a). Electron densities derived from the differential absorption experiment are shown in (b). In both (a) and (b) the circles represent 2 MHz measurements and the crosses represent 5 MHz measurements.

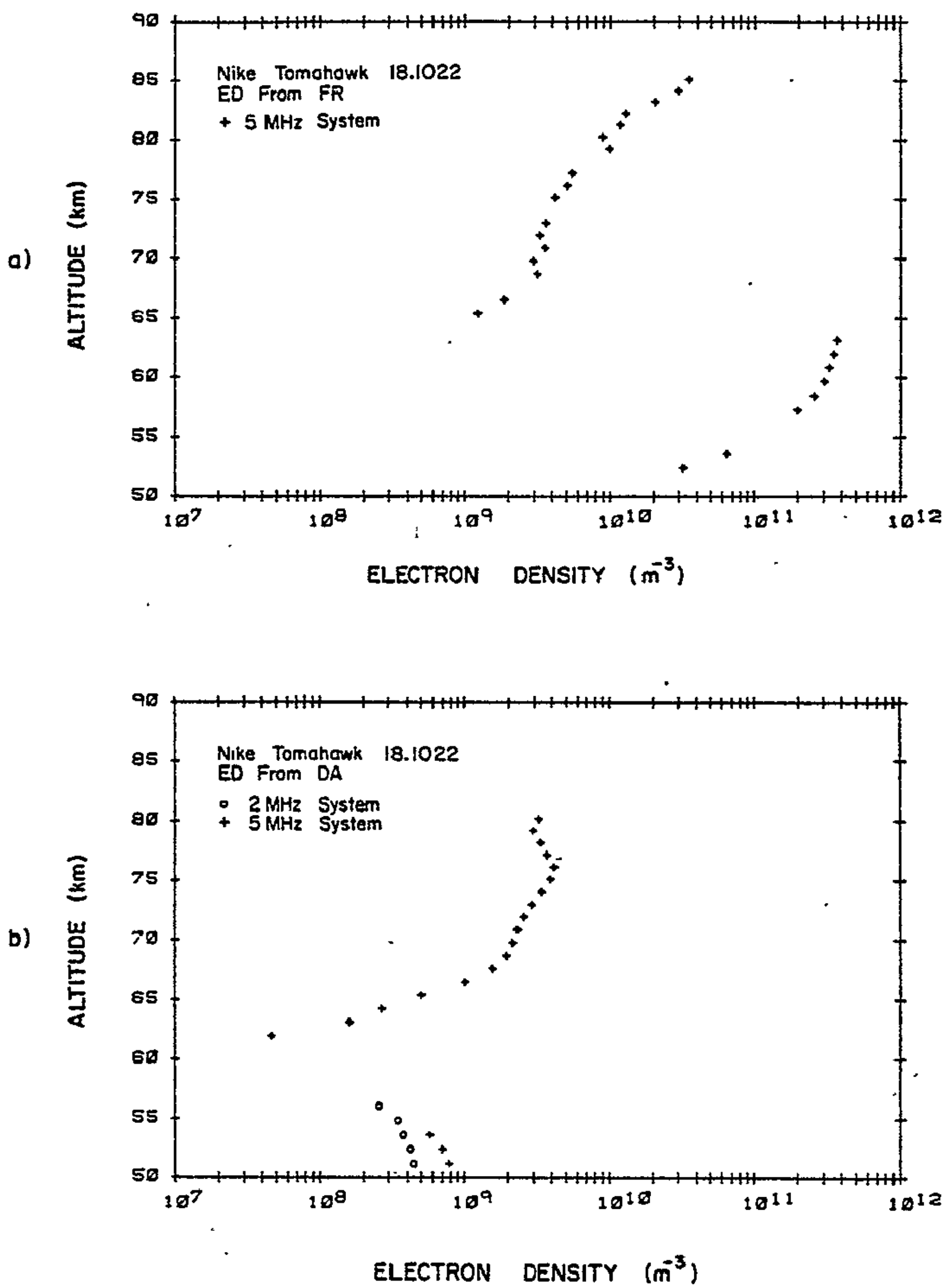


Figure 6.6 Electron density plots from Nike Tomahawk 18.1022, launched at 1054:10 LST (1654:10 UT) on 26 February 1979, during the eclipse. Electron densities derived from the Faraday rotation experiment are shown in (a). Electron densities derived from the differential absorption experiment are shown in (b). In (b) the circles represent the 2 MHz measurements and in both (a) and (b) the crosses represent the 5 MHz measurements. 2 MHz Faraday rotation data were unavailable.

Table 6.1a 18.1020 electron density from Faraday rotation.

TIME FROM LAUNCH (sec)	2 MHz		5 MHz	
	FARADAY RATE (deg/sec)	ELECTRON DENSITY m^{-3}	FARADAY RATE (deg/sec)	ELECTRON DENSITY m^{-3}
50	-2.6604	unmatchable	-3.8052	unmatchable
51	-5.5296	unmatchable	-7.5078	unmatchable
52	0.9108	1.444×10^9	1.4364	9.322×10^9
53	-0.1764	unmatchable	0.8784	6.633×10^9
54	3.6594	4.324×10^9	1.6380	1.654×10^{10}
55	-188.0478	1.359×10^{11}	75.0960	unmatchable
56	-1.9476	unmatchable	-1.2906	8.680×10^{10}
57	-4.8366	unmatchable	-4.1760	1.770×10^{11}
58	-3.0744	unmatchable	-2.0646	2.294×10^{11}
59	0.2466	2.329×10^8	-1.6434	2.743×10^{11}
60	-4.8654	unmatchable	0.8478	1.226×10^9
61	3.1752	3.989×10^9	1.3284	1.428×10^9
62	-3.1140	unmatchable	1.2366	1.028×10^9
63	-1.7478	9.436×10^9	1.5462	1.047×10^9
64	-10.7298	2.340×10^{10}	1.4418	8.388×10^8
65			12.9456	6.543×10^9
66			12.7350	5.713×10^9
67			10.1196	4.146×10^9
68			25.3242	9.602×10^9
69			76.0572	2.659×10^{10}
70			-7.0722	unmatchable

Table 6.1b 18.1020 electron density from differential absorption.

TIME FROM LAUNCH (sec)	2 MHz		5 MHz	
	DIFFERENTIAL ABSORPTION RATE (dB/sec)	ELECTRON DENSITY m^{-3}	DIFFERENTIAL ABSORPTION RATE (dB/sec)	ELECTRON DENSITY m^{-3}
50	--	--	--	--
51	-0.050	2.922×10^8	-0.0001*	7.638×10^5
52	-0.088	3.907×10^8	-0.0001*	6.168×10^5
53	-0.162	5.513×10^8	-0.025	1.264×10^8
54	-0.262	6.932×10^8	-0.125	5.282×10^8
55	-0.288	5.965×10^8	-0.188	6.728×10^8
56	-0.212	3.457×10^8	-0.112	unmatchable
57	-0.038	4.916×10^7	0.038	1.019×10^8
58	0.175	unmatchable	0.175	4.190×10^8
59	0.412	unmatchable	0.250	5.447×10^8
60	0.700	unmatchable	0.338	6.879×10^8
61	1.700	unmatchable	0.475	9.070×10^8
62	3.088	unmatchable	0.725	1.313×10^9
63			1.175	2.082×10^9
64			1.238	2.230×10^9
65			1.900	3.507×10^9

*experimentally zero; however -0.0001 is the minimum DA rate which can be safely entered into program DA2NE.

Table 6.2a 18.1021 electron density from Faraday rotation.

	2 MHz		5 MHz	
TIME FROM LAUNCH (sec)	FARADAY RATE (deg/sec)	ELECTRON DENSITY m^{-3}	FARADAY RATE (deg/sec)	ELECTRON DENSITY m^{-3}
50	-8.7192	unmatchable	-7.7508	unmatchable
51	-9.5688	unmatchable	-6.4044	unmatchable
52	25.0722	3.194×10^{10}	24.3252	1.269×10^{11}
53	0.0252	3.021×10^{7}	1.5192	1.417×10^{10}
54	-7.9902	unmatchable	-6.6708	7.394×10^{10}
55	-350.2098	2.042×10^{11}	25.5222	unmatchable
56	-18.9252	unmatchable	-17.7390	2.094×10^{11}
57	-5.4000	unmatchable	-6.0588	2.330×10^{11}
58	-8.9370	unmatchable	-9.3762	2.811×10^{11}
59	-12.0618	unmatchable	-9.3384	3.138×10^{11}
60	-3.1608	unmatchable	-1.9746	3.357×10^{11}
61	-1.8954	unmatchable	0.0810	6.722×10^{7}
62	-1.6920	9.104×10^{9}	0.0198	1.341×10^{7}
63	1.5678	1.872×10^{9}	2.5794	1.498×10^{9}
64	1.2798	6.993×10^{8}	2.7288	1.380×10^{9}
65	4.3704	1.445×10^{9}	5.8662	2.578×10^{9}
66	5.6592	1.245×10^{9}	3.5496	1.401×10^{9}
67	9.9576	1.614×10^{9}	10.1322	3.732×10^{9}
68	-1.9638	4.038×10^{10}	-4.2930	unmatchable
69			-1.5210	unmatchable
70			-9.0468	unmatchable
71			15.1416	4.807×10^{9}
72			18.4680	5.716×10^{9}
73			15.2676	4.672×10^{9}
74			24.3072	7.389×10^{9}
75			26.6166	8.214×10^{9}
76			65.2914	2.012×10^{10}
77			60.0300	1.854×10^{10}
78			35.8884	1.109×10^{10}
79			73.5480	2.238×10^{10}
80			99.8604	3.047×10^{10}
81			124.6104	3.756×10^{10}

Table 6.2b 18.1021 electron density from differential absorption.

	2 MHz		5 MHz	
TIME FROM LAUNCH (sec)	DIFFERENTIAL ABSORPTION RATE (deg/sec)	ELECTRON DENSITY m^{-3}	DIFFERENTIAL ABSORPTION RATE (deg/sec)	ELECTRON DENSITY m^{-3}
50	-0.012	7.363×10^7	-0.025	1.982×10^8
51	-0.012	5.569×10^7	-0.025	1.593×10^8
52	-0.012	4.209×10^7	-0.050	2.574×10^8
53	-0.025	6.742×10^7	-0.075	3.193×10^8
54	-0.125	2.633×10^8	-0.088	3.167×10^8
55	-0.225	3.718×10^8	-0.100	unmatchable
56	-0.212	2.748×10^8	-0.125	unmatchable
57	-0.138	1.435×10^8	-0.138	unmatchable
58	-0.038	3.242×10^7	-0.125	unmatchable
59	0.062	unmatchable	-0.100	unmatchable
60	0.188	unmatchable	-0.025	unmatchable
61	0.412	unmatchable	0.050	8.994×10^7
62	0.875	3.648×10^8	0.112	1.979×10^8
63	1.925	7.065×10^8	0.212	3.813×10^8
64	3.338	1.078×10^9	0.412	7.606×10^8
65	3.538	9.963×10^8	0.750	1.411×10^9
66			1.262	2.486×10^9
67			1.500	3.225×10^9
68			1.338	3.240×10^9
69			1.275	3.504×10^9
70			1.150	3.532×10^9
71			1.088	3.740×10^9
72			1.125	4.408×10^9
73			0.812	3.698×10^9
74			0.500	2.683×10^9
75			0.425	2.719×10^9

Table 6.3a 18.1022 electron density from Faraday rotation.

	5 MHz	
TIME FROM LAUNCH (sec)	FARADAY RATE (deg/sec)	ELECTRON DENSITY m^{-3}
50	-5.2056	unmatchable
51	-9.0540	unmatchable
52	-7.7580	unmatchable
53	4.2444	3.182×10^{10}
54	7.0848	6.397×10^{10}
55	151.1442	unmatchable
56	unavailable	unavailable
57	-1.4004	1.963×10^{11}
58	-4.997	2.565×10^{11}
59	-8.4780	2.988×10^{11}
60	-4.9374	3.261×10^{11}
61	-2.2464	3.487×10^{11}
62	-1.8504	3.676×10^{11}
63	-0.2772	unmatchable
64	2.3688	1.218×10^{9}
65	3.9834	1.863×10^{9}
66	-0.0126	unmatchable
67	7.9794	3.135×10^{9}
68	8.1306	2.939×10^{9}
69	10.3014	3.552×10^{9}
70	9.7146	3.257×10^{9}
71	11.1078	3.591×10^{9}
72	unavailable	unavailable
73	13.3182	4.162×10^{9}
74	16.1874	5.029×10^{9}
75	17.7390	5.424×10^{9}
76	unavailable	unavailable
77	31.6584	9.829×10^{9}
78	29.0322	8.772×10^{9}
79	36.9936	1.166×10^{10}
80	41.0580	1.277×10^{10}
81	64.8810	2.027×10^{10}
82	95.6016	2.953×10^{10}
83	117.3078	3.481×10^{10}

The data for 2 MHz Faraday rate and electron density was not available.

Table 6.3b 18.1022 electron density from differential absorption.

	2 MHz		5 MHz	
TIME FROM LAUNCH (sec)	DIFFERENTIAL ABSORPTION RATE (deg/sec)	ELECTRON DENSITY m^{-3}	DIFFERENTIAL ABSORPTION RATE (deg/sec)	ELECTRON DENSITY m^{-3}
50	-0.075	5.143×10^8	-0.112	9.676×10^8
51	-0.100	5.257×10^8	-0.112	7.862×10^8
52	-0.112	4.443×10^8	-0.138	7.791×10^8
53	-0.138	4.207×10^8	-0.150	6.977×10^8
54	-0.162	3.744×10^8	-0.150	5.711×10^8
55	-0.188	3.422×10^8	-0.150	unmatchable
56	-0.175	2.546×10^8	-0.150	unmatchable
57	-0.125	1.460×10^8	-0.138	unmatchable
58	-0.062	5.747×10^7	-0.112	unmatchable
59	0.012	unmatchable	-0.100	unmatchable
60	0.088	unmatchable	-0.062	unmatchable
61	0.162	unmatchable	0.025	4.584×10^7
62	0.325	unmatchable	0.088	1.597×10^8
63	0.888	3.472×10^8	0.150	2.672×10^8
64	1.975	6.468×10^8	0.288	5.004×10^8
65	3.012	9.028×10^8	0.538	1.004×10^9
66	2.962	8.002×10^8	0.788	1.550×10^9
67			0.912	1.948×10^9
68			0.938	2.142×10^9
69			0.900	2.297×10^9
70			0.875	2.532×10^9
71			0.888	2.889×10^9
72			0.912	3.378×10^9
73			0.912	3.897×10^9
74			0.825	4.112×10^9
75			0.638	3.677×10^9
76			0.475	3.310×10^9
77			0.362	2.918×10^9
78			0.350	3.215×10^9

electron density at the highest altitudes. This is due to polarization errors in the transmitted signals (GINTHER AND SMITH, 1975). As the ratio of the ordinary to the extraordinary signal is being kept constant, the magnitude of the error in the ordinary signal continues to increase. Eventually the magnitudes of the error in the ordinary wave and of the extra-ordinary wave at the rocket become equal. At this point, the servo system begins following the signal generated by the ordinary wave and its error signal rather than the signal generated by the ordinary and extra-ordinary waves.

6.2.2 Determination of electron density. A matching technique is used by programs FR2NE and DA2NE to find the electron density which corresponds to a particular rotation or absorption rate. In order to view how the programs match a rate, rotation and absorption rates have been plotted versus electron density. Figures 6.7 and 6.8 indicate the variation of Faraday rotation rate with electron density at different altitudes. The variation of the differential absorption rate with electron density and with height is shown in Figures 6.9 and 6.10. Trajectory data from 18.1020 provides a realistic basis for these curves.

These curves indicate that for some rates there are two electron densities. For this reason the scan of electron densities by programs FR2NE and DA2NE begins at 1 m^{-3} and increments until an invertal containing the rate to be matched is found.

The experimental rates in Tables 6.1 through 6.3 which were unmatchable usually fell in a discontinuity of the theoretical rate (represented by dotted lines in Figures 6.7 through 6.10). The other rates were unmatchable because the theory did not predict any rates of that sign (i.e., positive or negative).

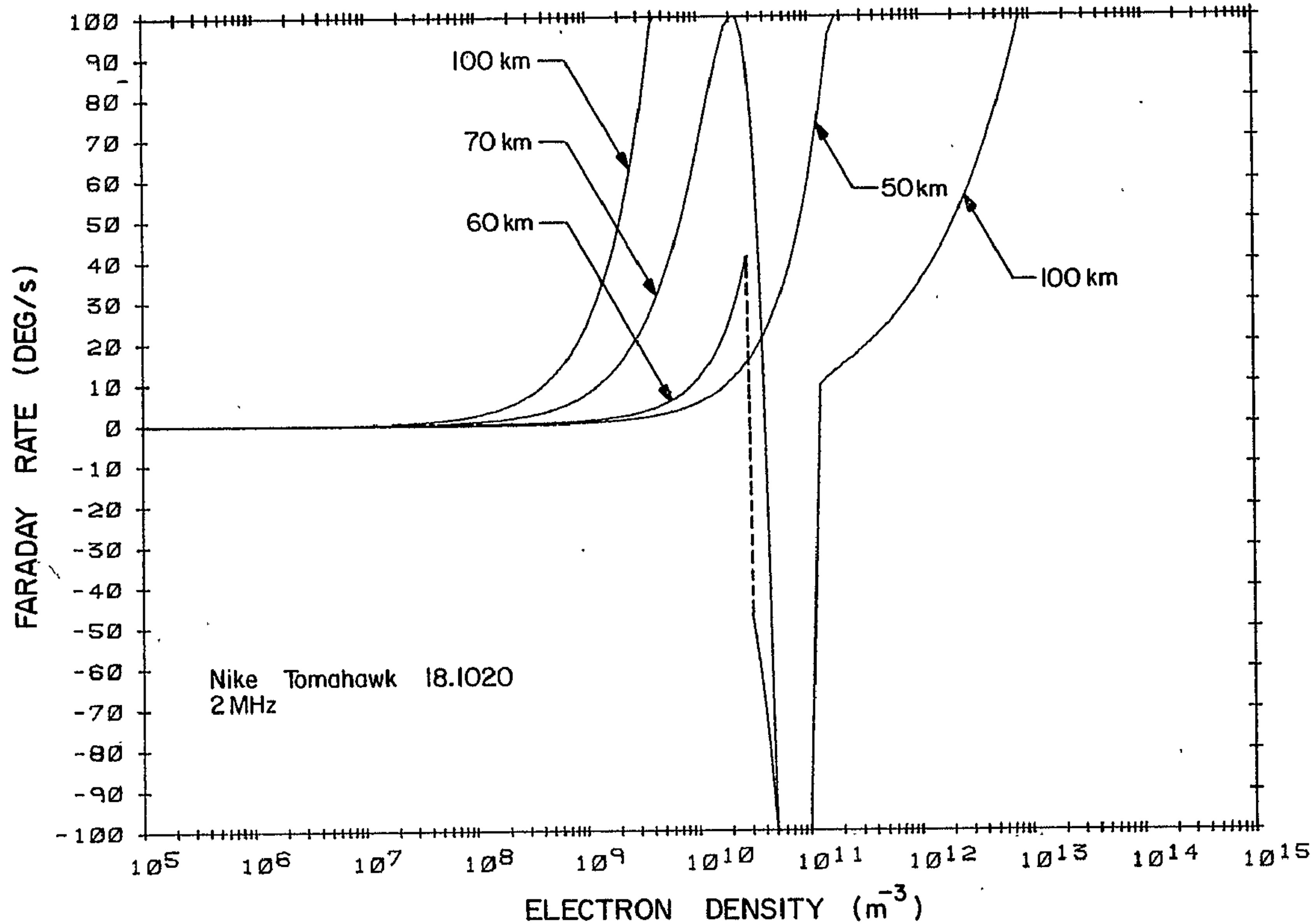


Figure 6.7 Plots of calculated 2 MHz Faraday rotation rates versus electron density at several altitudes. The dashes represent a discontinuity in the Faraday rotation rate. Trajectory data from Nike Tomahawk 18.1020 were input into a modified FR2NE program in order to generate these Faraday rotation rates.

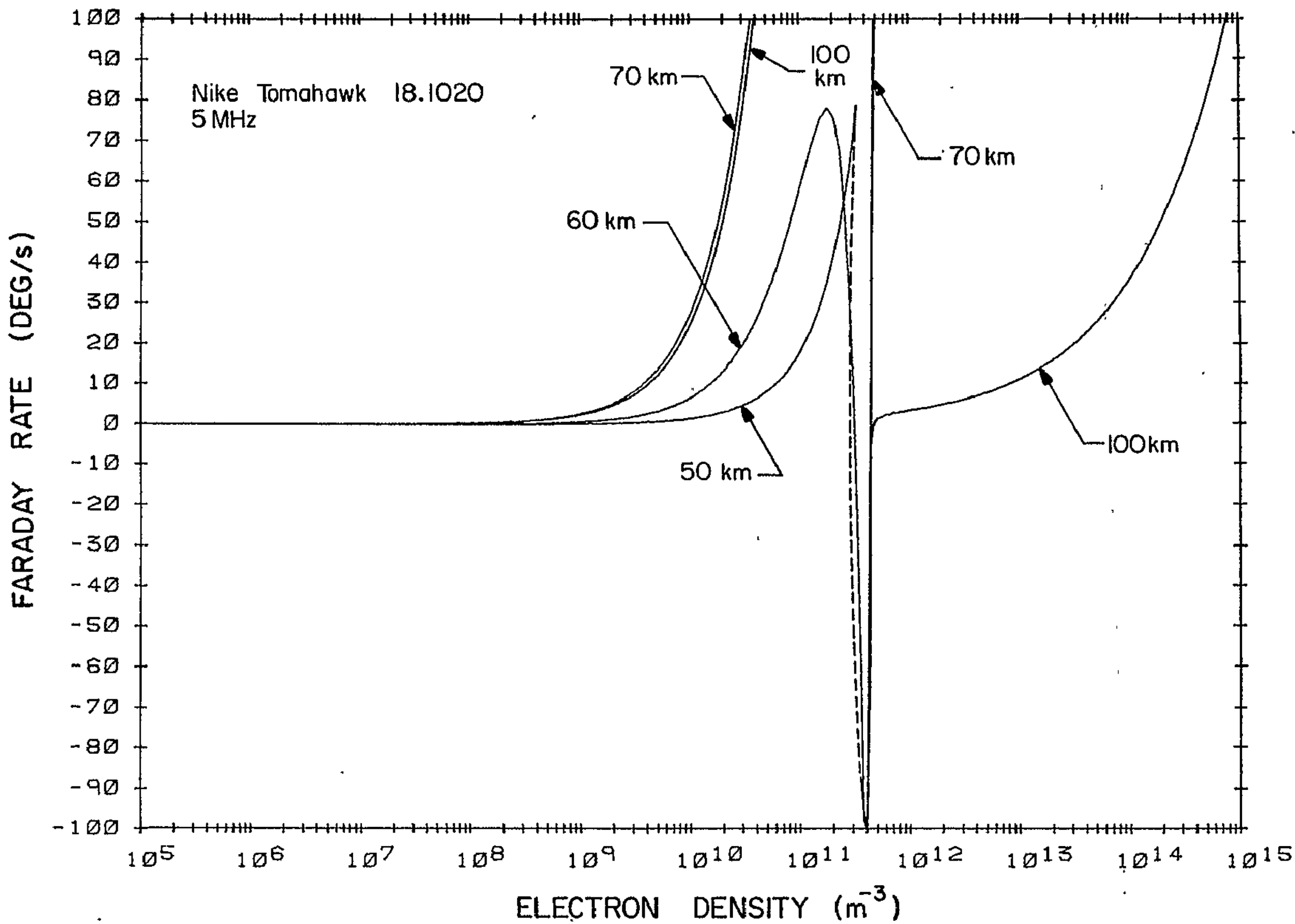


Figure 6.8 Plots of calculated 5 MHz Faraday rotation rates versus electron density at several altitudes. The dashes represent a discontinuity in the Faraday rotation rate. Trajectory data from Nike Tomahawk 18.1020 were input into a modified FR2NE program in order to generate these Faraday rotation rates.

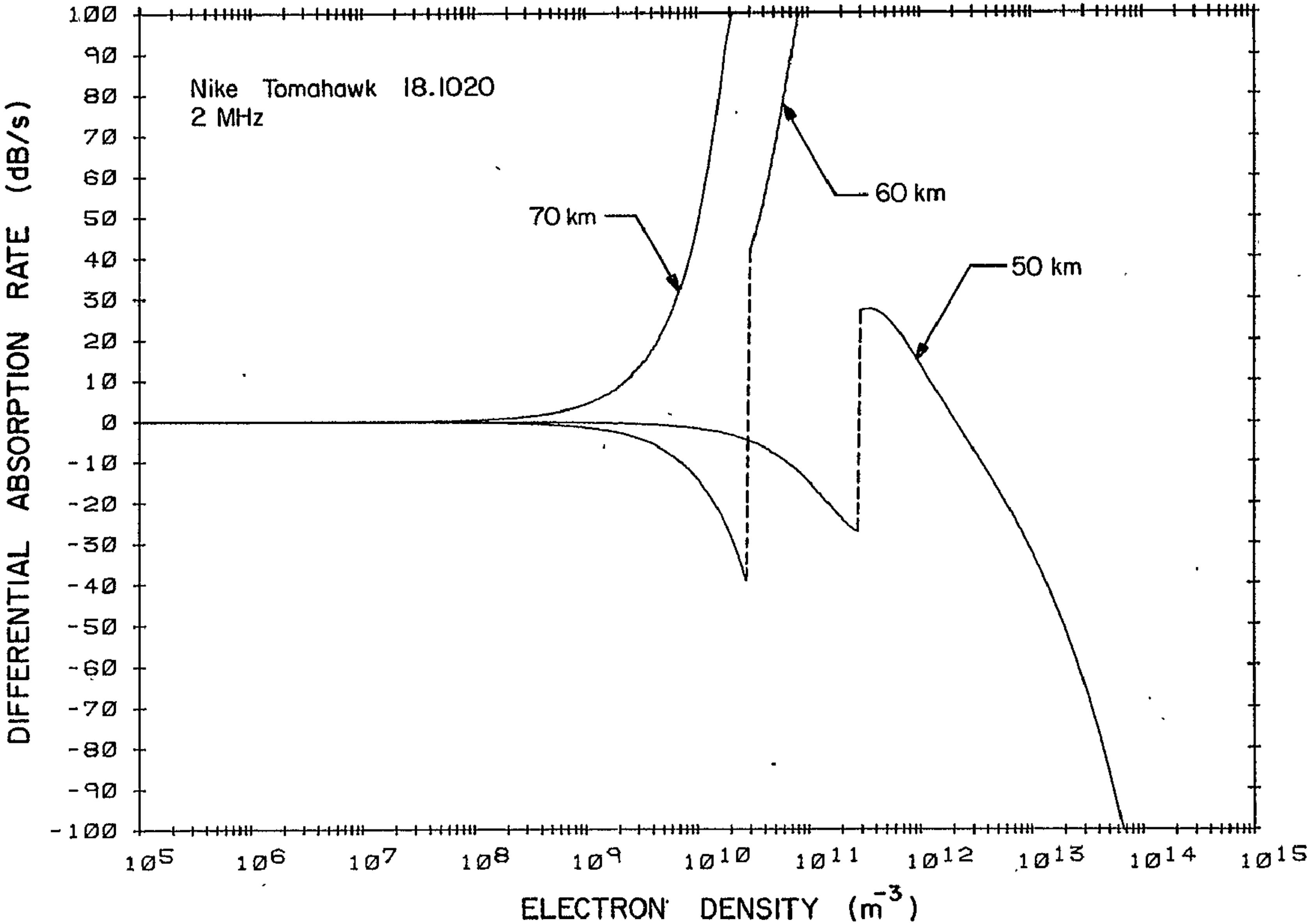


Figure 6.9 Plots of calculated 2 MHz differential absorption rates versus electron density at several altitudes. The dashes represent discontinuities in the differential absorption rates. Trajectory data from Nike Tomahawk 18.1020 were input into a modified DA2NE program in order to generate these differential absorption rates.

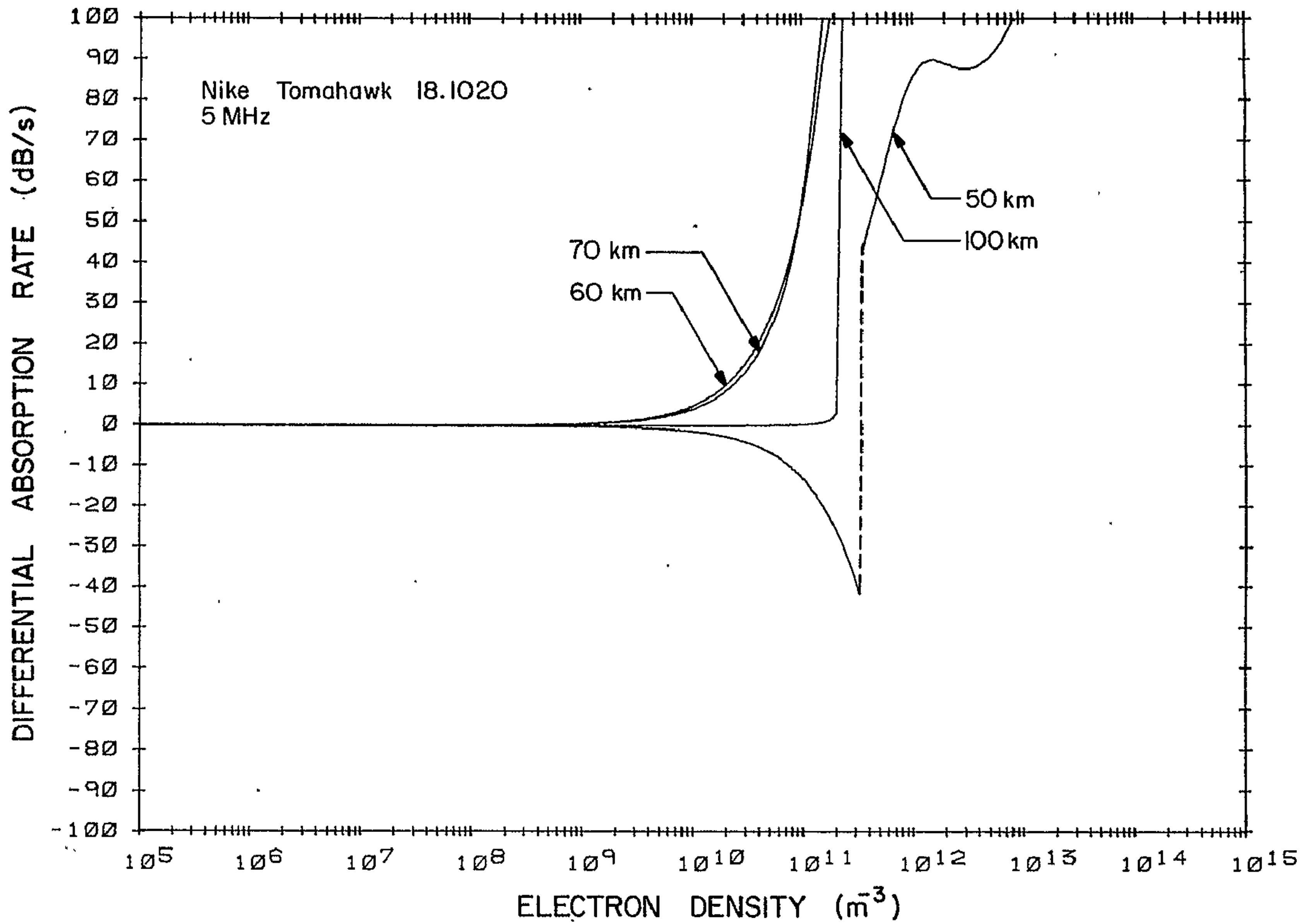


Figure 6.10 Plots of calculated 5 MHz differential absorption rates versus electron density at several altitudes. The dashes represent a discontinuity in the differential absorption rate. Trajectory data from Nike Tomahawk 18.1020 were input into a modified DA2NE program in order to generate these differential absorption rates.

6.3 Electron Density Profiles

The electron density profiles for 18.1020, 18.1021 and 18.1022 are shown in Figures 6.11, 6.12 and 6.13, respectively. The ascent portion of the profiles is represented by the solid line while the descent portion is represented by the dashed lines. Due to the inconsistencies of the electron densities derived from the Faraday rotation experiment, the electron densities derived from the differential absorption experiment were used to calibrate the probe profiles. Table 6.4 lists the probe current calibration factors and the time periods over which they were used. The first calibration value is used until payload separation and the second value after separation (see Figures 6.1 through 6.3). Although the calibration factors are established over a limited height range for these flights, the values are in close agreement with calibration factors established for similar flights at Wallops Island.

The electron density profile of the pre-eclipse launch, 18.1020 (Figure 6.11), shows the E-region electron density is essentially constant at about 1×10^5 cm^{-3}.

The electron density profile of the first launch in totality, 18.1021 (Figure 6.12), shows a difference in electron density between ascent and descent which can not be attributed to experimental effects. This difference indicates temporal and/or spatial variations of electron density.

For 18.1022 (Figure 6.13) the electron densities are comparable to the previous launch in totality (18.1021) but the difference between ascending and descending electron densities is not as pronounced.

6.4 Energetic Particle Profiles

The two solid-state particle detectors used in the energetic particle experiment were identical except for the thickness of the aluminum surface

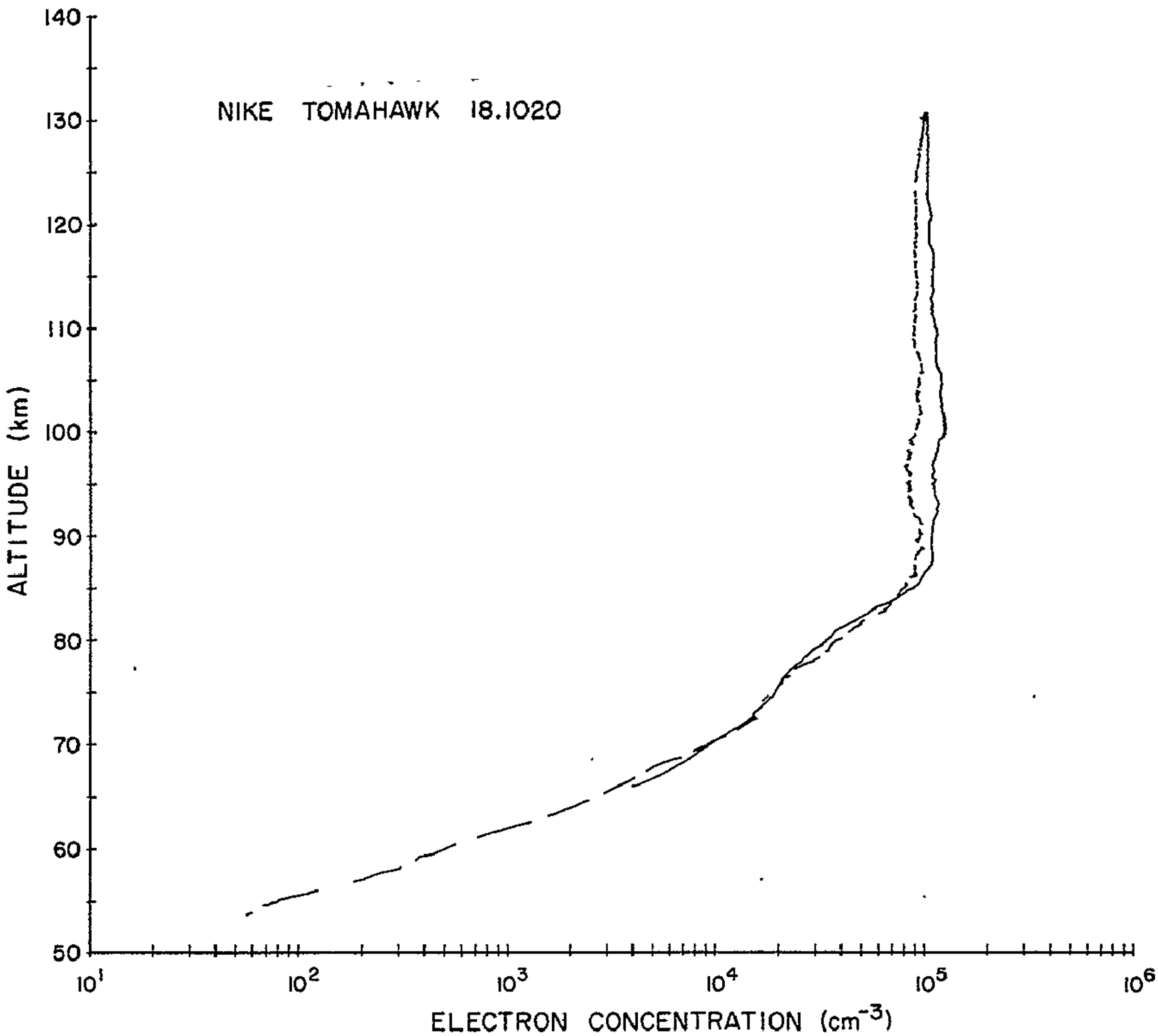


Figure 6.11 Electron density profile from Nike Tomahawk 18.1020, launched at 1652 UT on 24 February 1979. Ascent: solid curve; descent: dashed curve.

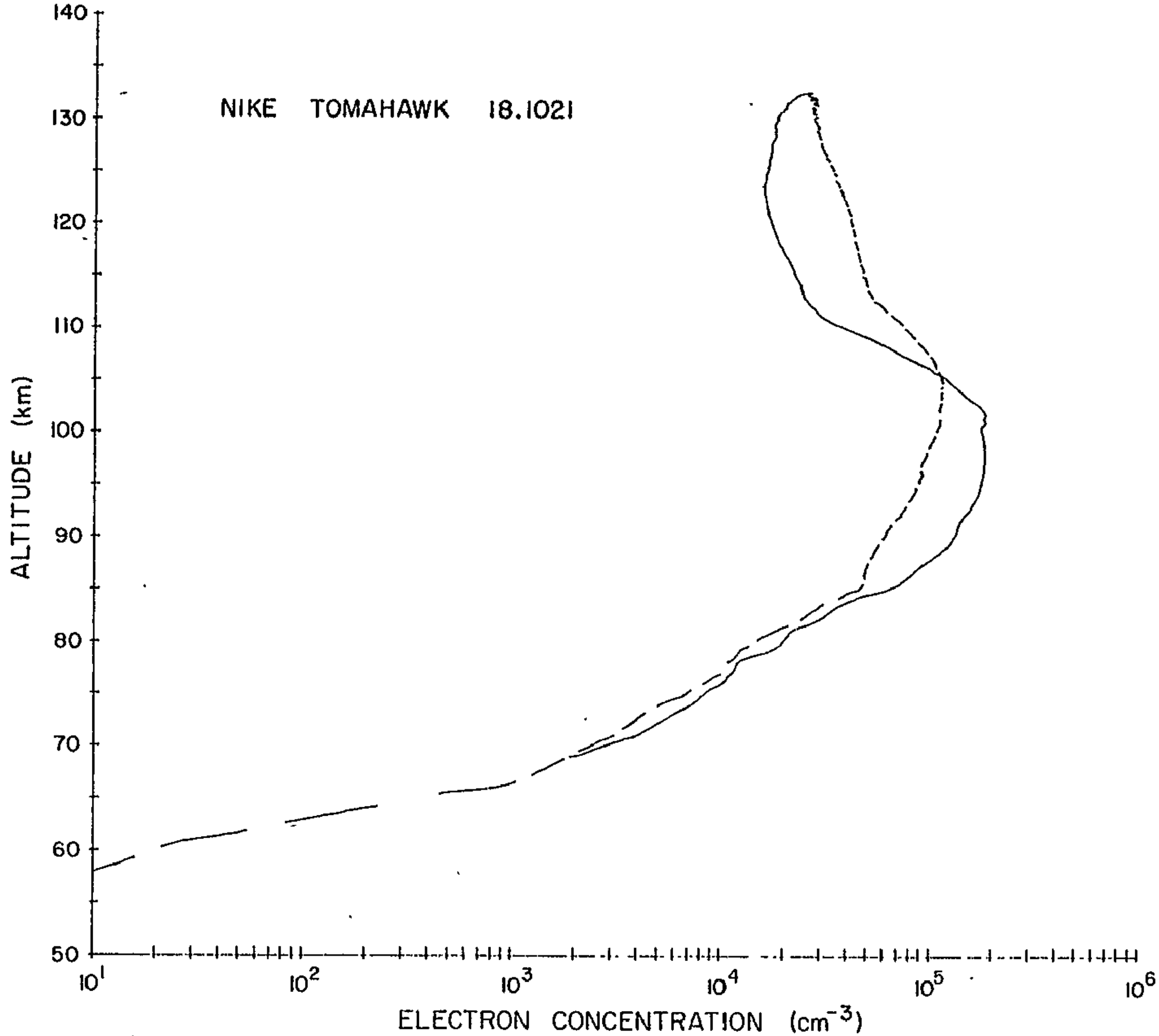


Figure 6.12 Electron density profile from Nike Tomahawk 18.1021, launched at 1654:10 UT on 26 February 1979, during the eclipse. Ascent: solid curve; descent: dashed curve.

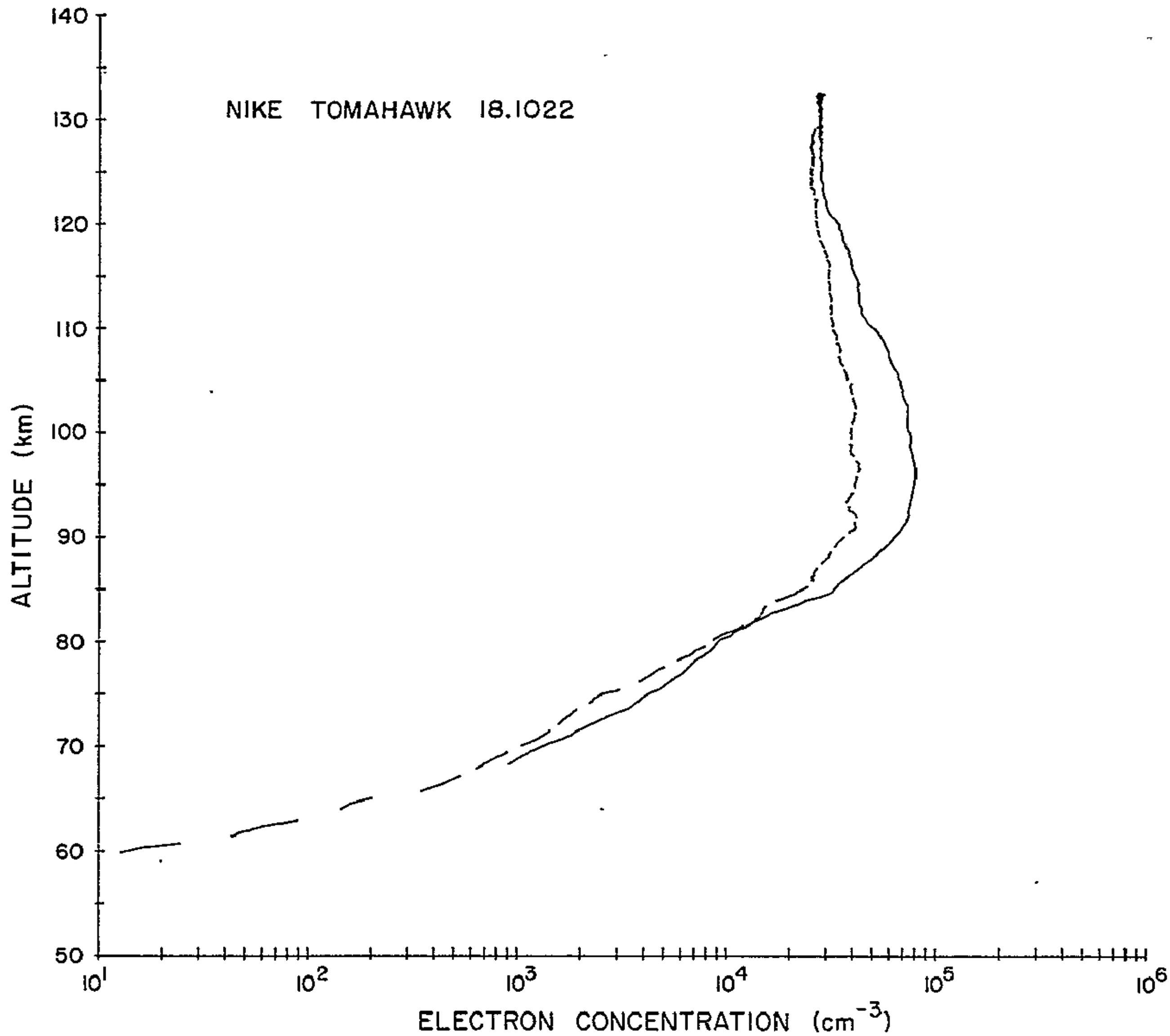


Figure 6.13 Electron density profile from Nike Tomahawk 18.1022, launched at 1654:10 UT on 26 February 1979, during the eclipse. Ascent: solid curve; descent: dashed curve.

Table 6.4 Probe current calibration factors.

Flight	N/I(cm^{-3} A^{-1})	Time from launch (sec) over which N/I value is used
18.1020	1.6×10^{10}	65 to 291
	2.0×10^{10}	291 to 310
18.1021	1.6×10^{10}	66 to 291
	2.5×10^{10}	291 to 320
18.1022	1.56×10^{10}	66 to 293
	2.0×10^{10}	293 to 314

layer. A detector with 40 μg cm^{-2} of aluminum was used during rocket ascent. For descent a detector with 100 μg cm^{-2} of aluminum was used. The ratio of the fluxes in the two detectors allows particle identification.

Count rates for energetic particles versus time are given in Figures 6.14, 6.15 and 6.16 for 18.1020, 18.1021 and 18.1022, respectively. The particles consist mainly of electrons.

Figure 6.15 shows the particle data for 18.1021. Apogee occurred at approximately 180 sec after launch. At that time the experiment was switched to the detector with the thicker metal layer. If many particles heavier than electrons (e.g., H^+, He^+, O^+) were present there would have been a significant difference in the count rate, which was not observed.

All three flights show high count rates, indicating an auroral event on both days. The eclipse data, however, show unusually large fluctuations in the count rate, indicating a pulsating aurora (SMITH ET AL., 1980). An interesting feature of Figures 6.14 and 6.15 is that the fluctuations are more pronounced at the higher energies.

The count rates for 18.1021 and 18.1022 have been examined for correlations. A "scatter plot" of particle count rates (E > 70 keV) for the two rockets is shown in Figure 6.17. For each second between 1655:39 UT and 1656:39 UT the count rate from 18.1022 is plotted against the count rate from 18.1021. If there is a temporal correlation the points will lie on a straight line. Since the points are scattered there is no temporal correlation (over one second) between the particles observed on the two totality flights. However, examination of the data on a time scale of much less than one second does show occasional "micro-bursts" occurring simultaneously on the two rockets.

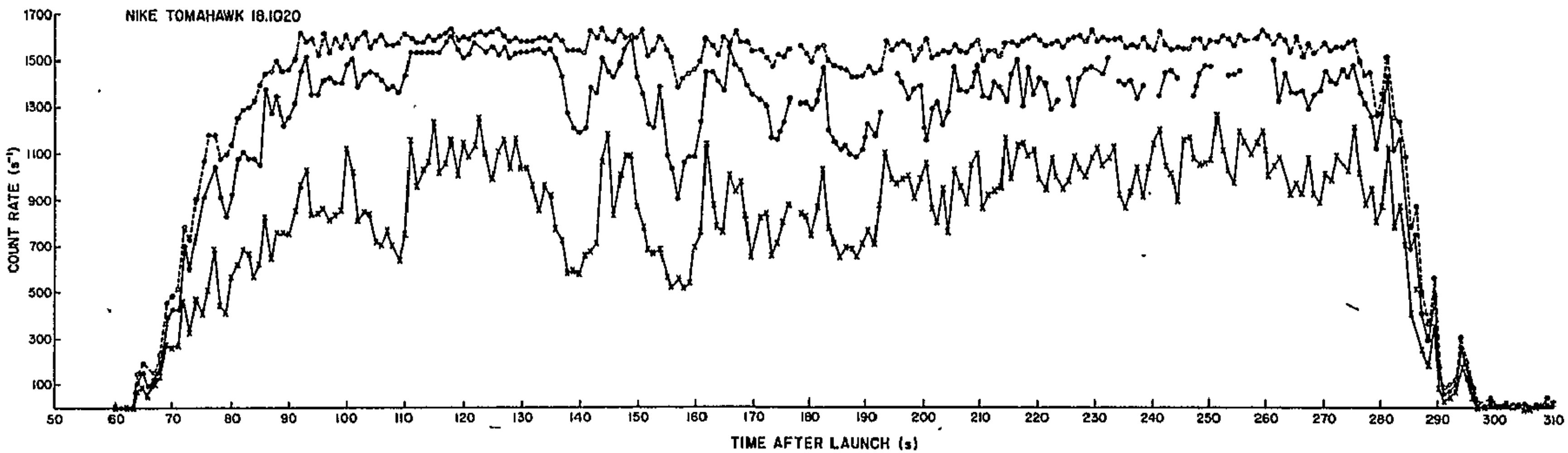


Figure 6.14 Count rates for energetic particles from Nike Tomahawk 18.1020, launched at 1652 UT on 24 February 1979.

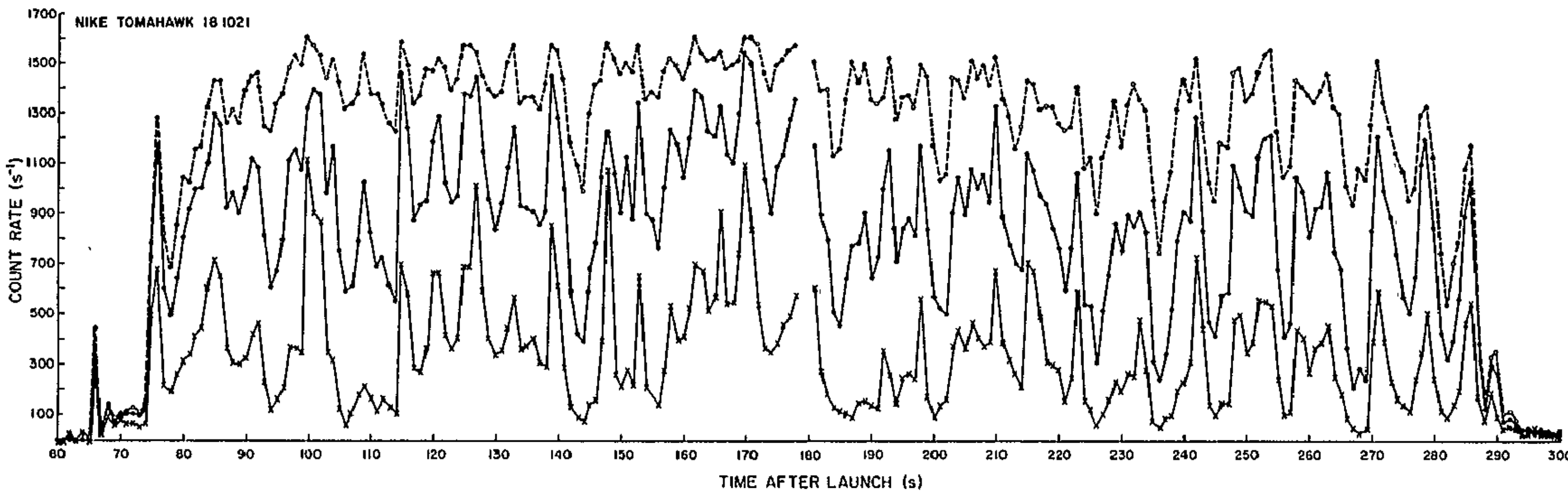


Figure 6.15 Count rates for energetic particles from Nike Tomahawk 18.1021, launched at 1652 UT on 26 February 1979, during the eclipse.

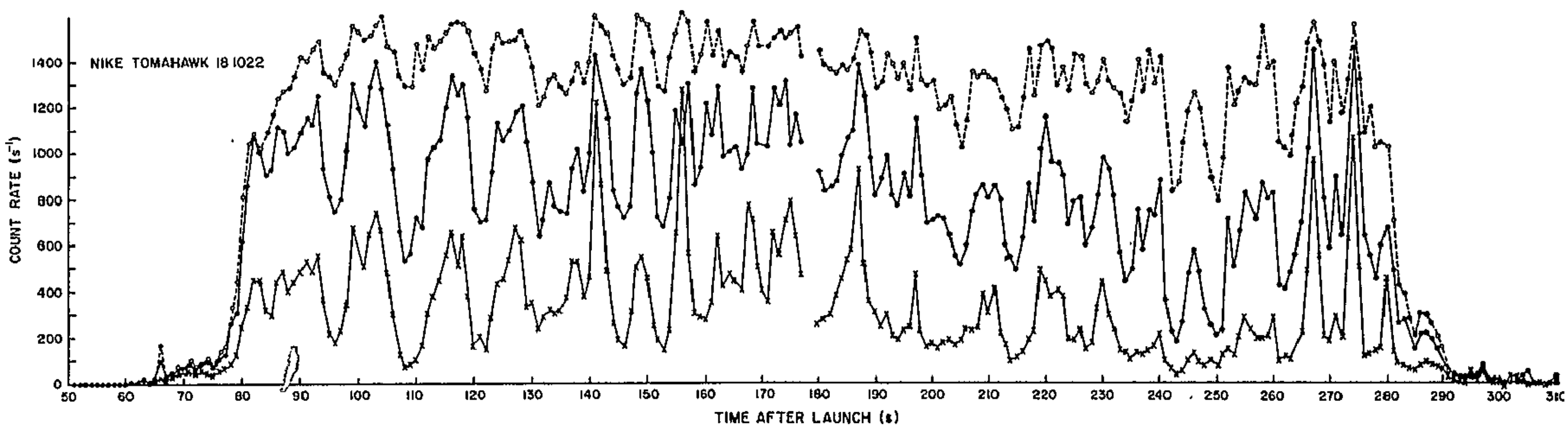


Figure 6.16 Count rates for energetic particles from Nike Tomahawk 18.1022, launched at 1654:10 UT on 26 February 1979, during the eclipse.

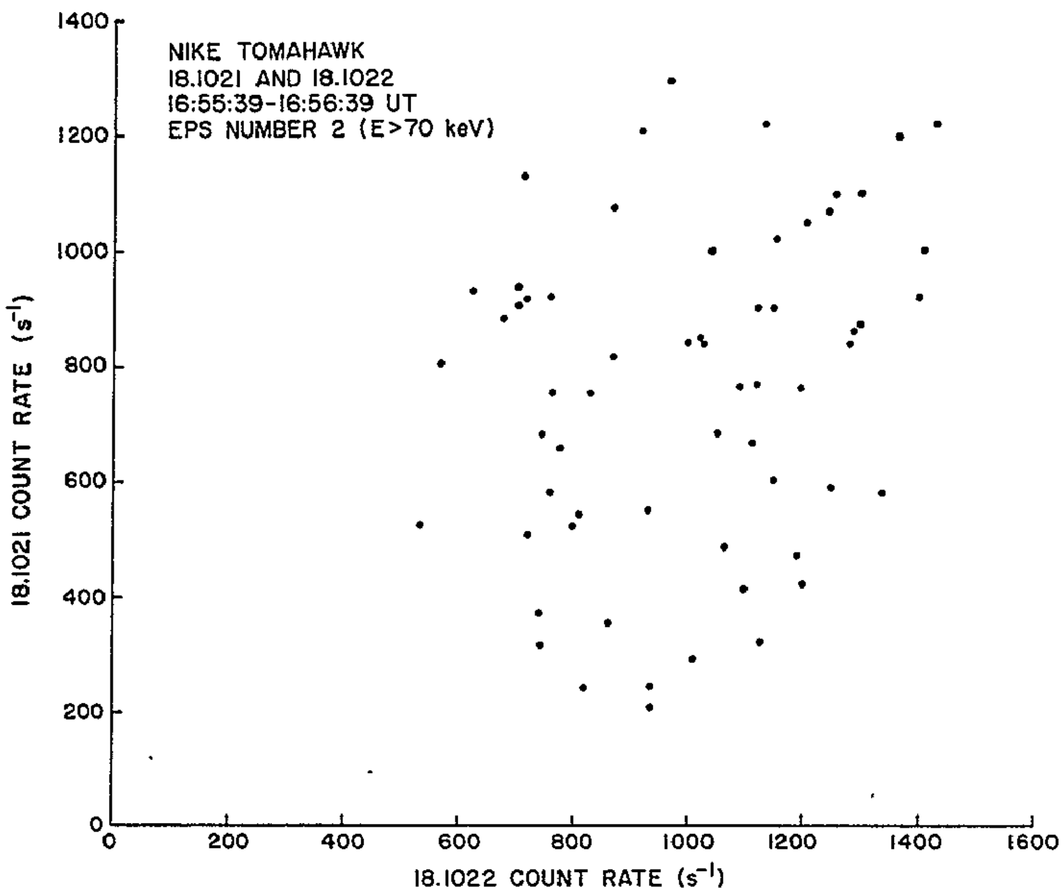


Figure 6.17 Count rates of energetic-particle sensor number 2 (E > 70 keV) for 18.1021 launched at 1652 UT 26 February 1979 versus count rate of energetic-particle number 2 (E > 70 keV) for 18.1022 launched at 1654:10 UT 26 February 1979 at one-second intervals between 1655:39 and 1656:39 UT.

An examination of the spatial correlation of the particle flux was done also. A plot of the simultaneous values of the two rockets (as above), separated horizontally by about 38 km, showed the count rates to be uncorrelated.

6.5 Discussion

The electron density profile from the pre-eclipse flight is shown again in Figure 6.18, this time along with a previous full-sun profile, from Nike Apache 14.435 launched at Wallops Island for the March 1970 eclipse.

The electron densities above 110 km agree. Below 110 km the 18.1020 electron density profile is enhanced by an order of magnitude in the D-region. This enhancement may be qualitatively explained by the large flux of electrons observed by the same rocket.

During totality, on 26 February, the electron density above 110 km (see Figure 6.19) is reduced by a factor of about three, as expected for the photochemical equilibrium model (section 2.4). Below 110 km, however, the electron density is much greater than that observed during previous eclipses (Nike Apache 14.436, Wallops Island, 1970, for example). Again this can be attributed to the additional ionization due to energetic particles.

Figure 6.20 shows the particle fluxes measured on the 24th and 26th vs. altitude. The flux on the 26th is much less than that measured on the 24th at altitudes below 80 km. The particle flux on the 26th does show a greater variability above 80 km, particularly at the higher energies ($E > 120$ keV).

The average count rate from 120 sec after launch to 169 sec (110 km to 130 km) for the three flights is shown in Figure 6.21. It can be noted that the average count rates for a threshold energy of 40 keV are nearly equal for all three flights. At threshold energies of 70 and 120 keV, however, the average count rate on the 24th (18.1020) is significantly greater than

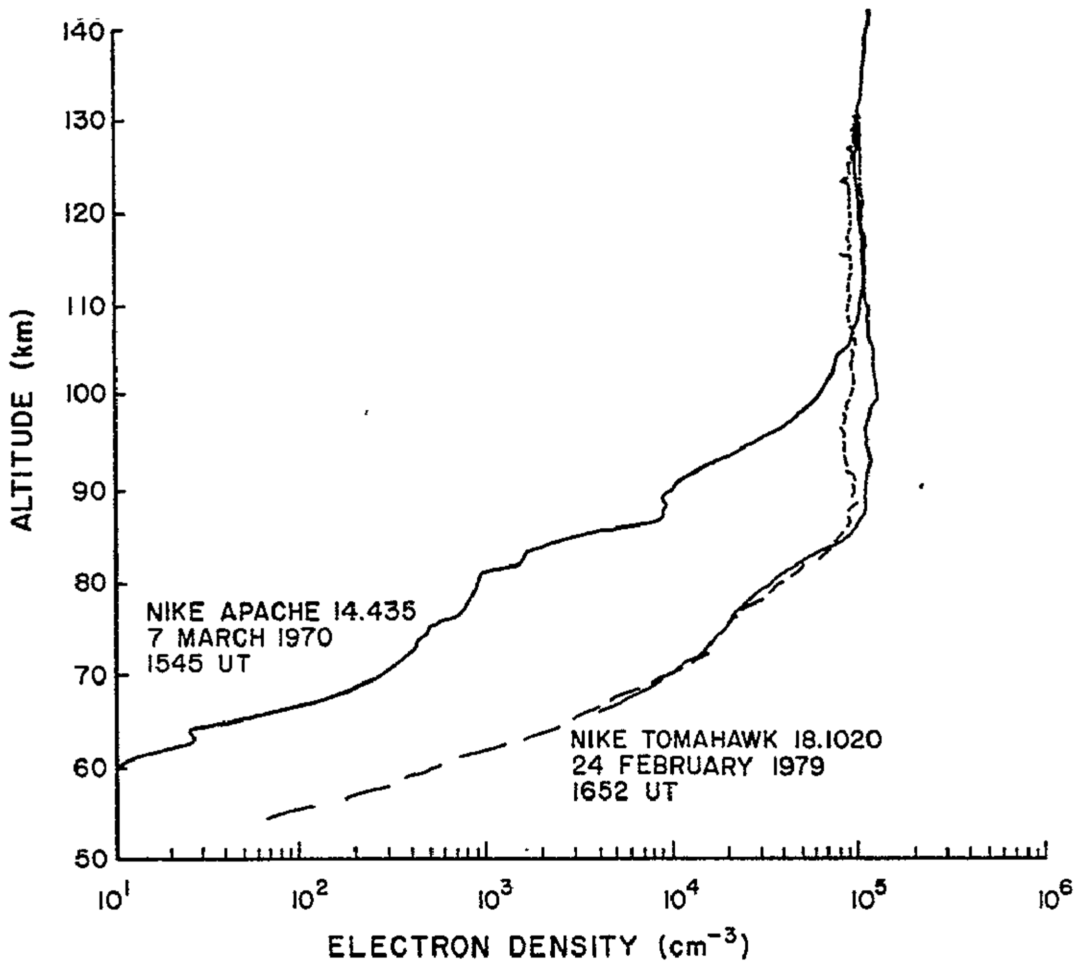


Figure 6.18 Electron density profiles for two pre-eclipse rockets.

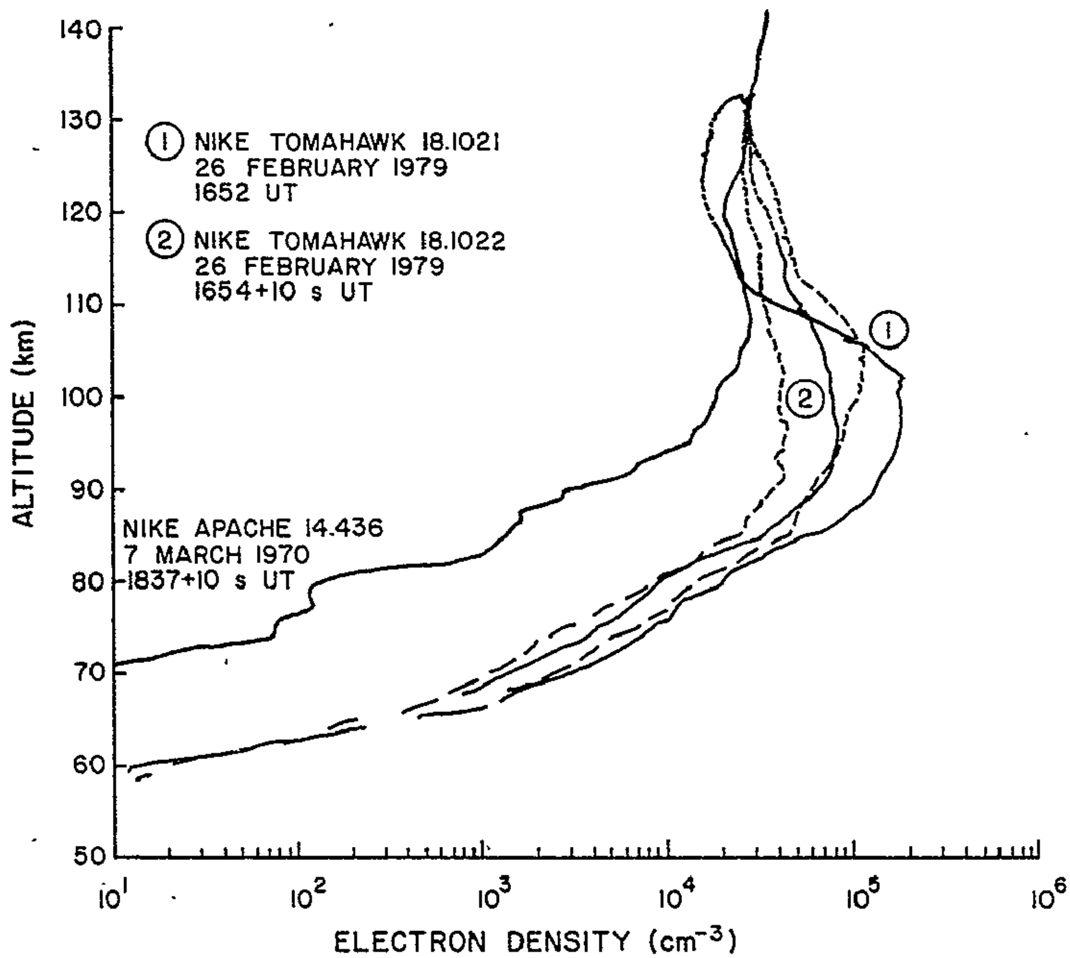


Figure 6.19 Electron density profiles for three rockets launched during eclipse totality.

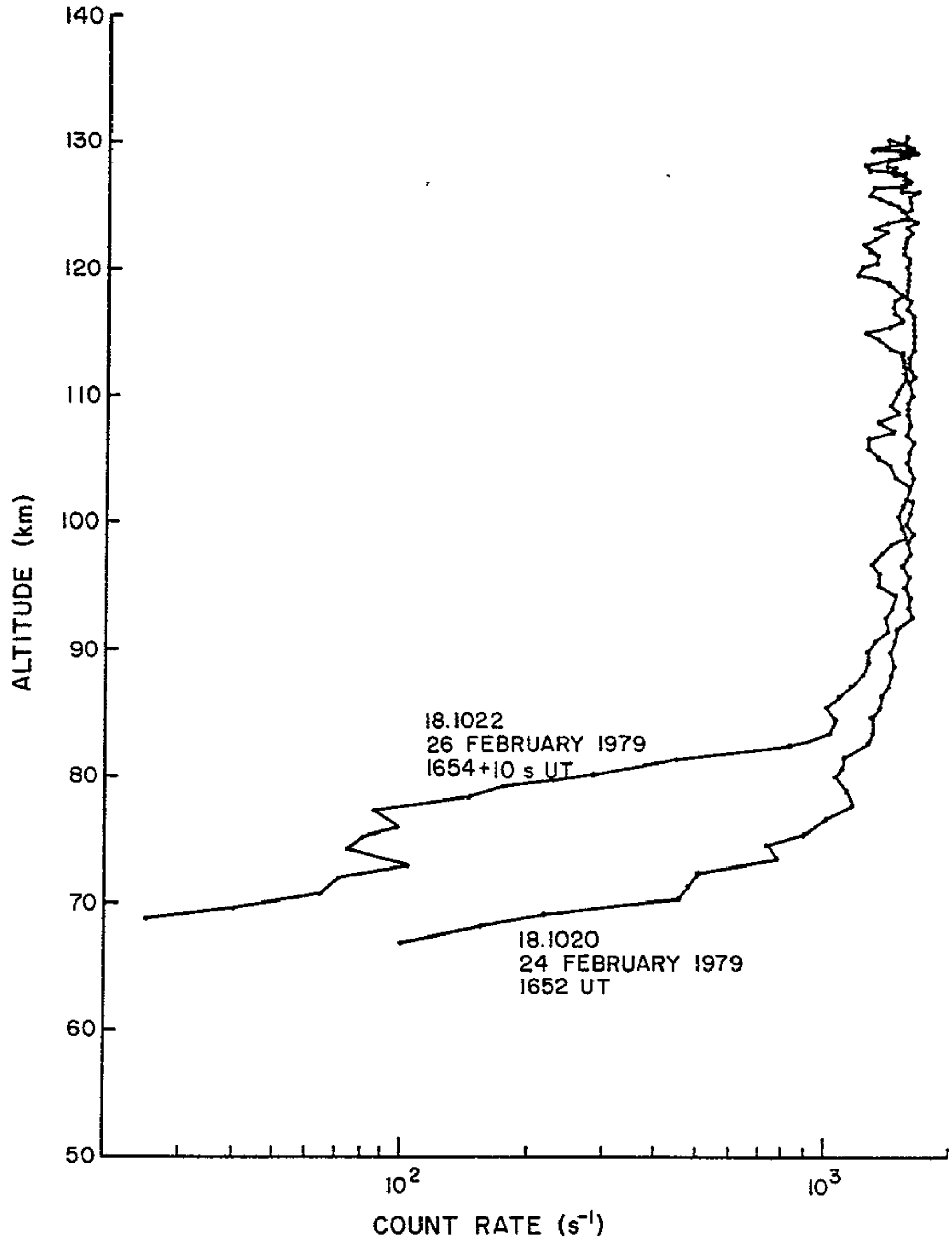


Figure 6.20. Count rates for energetic particle (>40 keV) sensors for Nike Tomahawk 18.1020 (pre-eclipse) and Nike Tomahawk 18.1022 (eclipse totality).

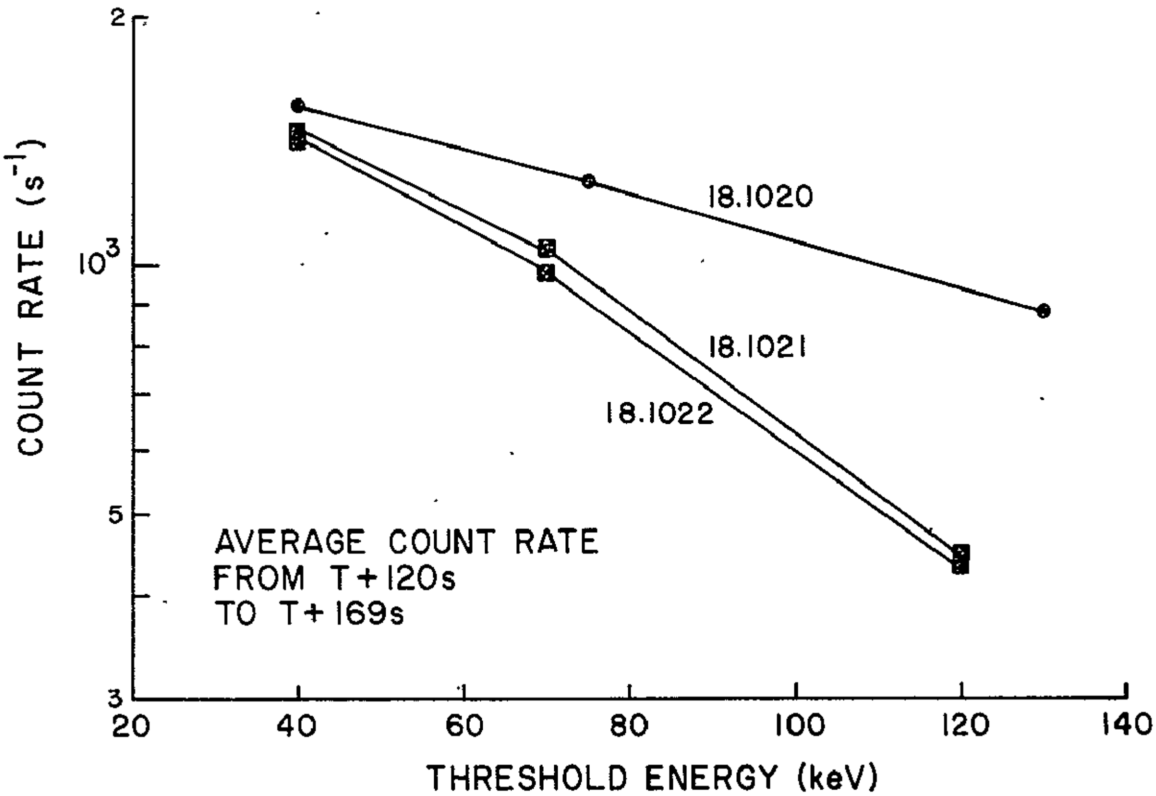


Figure 6.21 Average count rates plotted against threshold energy from T + 120 s to T + 169 s UT (110 km to apogee) for the three rockets of the 1979 eclipse operation.

those for the two eclipse flights (18.1021 and 18.1022). This is consistent with the high count rate observed below 80 km on the 24th (Figure 6.20; higher energy particles penetrate deeper into the D-region).

Finally, a simple exercise which can be performed given an electron density profile is an estimation of the production rate, q. Rewriting Equation 2.19 gives

$$q = (1 + \lambda)(\alpha_d + \lambda\alpha_i)N_e^{\,2} \tag{6.1}$$

where α_d is the dissociative recombination coefficient and α_i is the ion-ion recombination coefficient.

Recall from Equation 2.16 that

$$\lambda = \frac{N_-}{N_e} = \frac{a\,N^2}{(\rho + \delta N_m + \alpha_i\,N_+)} \tag{6.2}$$

which can be approximated by

$$\lambda = \frac{A}{P + C} \tag{6.3}$$

where A is an attachment rate and P and C are photodetachment and collisional detachment rates, respectively. λ is thus a function of altitude and intensity of photodetaching radiation, but not of N_e or N_-. Given

$$\begin{aligned} \alpha_d &= 5 \times 10^{-7}\ \text{cm}^3\ \text{s}^{-1} \\ \alpha_i &= 2 \times 10^{-7}\ \text{cm}^3\ \text{s}^{-1} \end{aligned} \tag{6.4}$$

and Rishbeth's model for λ (Figure 2.9) both q and N_- can be calculated from N_e.

18.1020 (full sun)

z (km)	λ	N_- (cm^{-3})	q ($cm^3 s^{-1}$)
55	15	1.2×10^3	0.4
65	2.25	6.8×10^3	28
75	0.15	3.0×10^3	244

18.1021 (eclipsed sun - no photodetachment)

z	λ	N_-	q
65	34	1.0×10^4	23
75	8	5.6×10^4	926

At 65 km the production rate for the no-sun situation is nearly equal to that of the full-sun situation while at 75 km the no-sun production rate is nearly four times the full-sun production rate. Figure 6.20 indicates the particle contribution to the ionization rate to be small at these altitudes. Based on photodetachment alone one would expect the no-sun production rate to be much lower than the full-sun production rates. Therefore, one can conclude that the detachment rate has been underestimated for totality, probably by the omission of associative detachment.

This conclusion is supported by the electron density profile of 18.1022. On ascent the rocket was in totality, while on descent 3% of the sun's disc was visible. Due to the increase of photodetaching radiation on descent, using the original ion production model one would expect an increase in electron density; however, the electron densities on ascent and descent do not differ in the D-region.

6.6 Recommendations for Further Study

The occurrence of a particle event during an eclipse provides a great opportunity for studying the effects of particle and electromagnetic radiation on the lower ionosphere. Data exist for daytime and nighttime auroras, but none for an intermediate condition such as an eclipse. A study of the data from flight 18.1022 may result in a better knowledge of the combined effects of electromagnetic and particle radiation on the lower ionosphere. Recall that rocket 18.1022 was in totality for only a portion of the flight.

The particle data can also be used to map the particle flux. Since rockets 18.1021 and 18.1022 were in the ionosphere simultaneously for a portion of their flights, the flux can be mapped spatially as well as temporally.

The observation of negative differential absorption, shown in Figures 5.12 and 5.13 poses experimental and theoretical questions concerning electromagnetic wave propagation in the lower D-region. The occurrence of negative Faraday rotation rates is consistent with magnetoionic theory and has been observed previously (for example, AIKIN ET AL., 1964) but it is not clear whether negative differential absorption can be explained by this theory.

APPENDIX I. Listing of IBM Fortran IV program WPROBE.

Included with the program is the JCL and input data necessary for execution. In this example probe data from flight 18.1020 is being processed. Probe current versus time data is routed to the Cyber 175 for further analysis.

```
//MCI JOB
/*ID PS=6899,NAME='M K MCINERNEY'
/*ID CODE=
/*ID REGION=125K
/*ID LINES=6500,TIME=2,IOREQ=1500
/*ID CARDS=2500
/*ID EJECT=YES
/*ID PUNCH=CYBER
/*ID NAME='PC20 (3KEMVUJ)'
/*SETUP UNIT=TAPE,R=D486,ID=(PFT203,NORING)
// EXEC FORTLDGO
//FORT.SYSIN DD *
C                        ----- PROGRAM WPROBE -----
C
C  THIS PROGRAM, WRITTEN FOR THE IBM 360, COMPUTES LANGMUIR PROBE CUR-
C  ENT FROM PROBE LOG VOLTAGE SAMPLES DIGITIZED AT WALLOPS ISLAND PCM
C  FACILITY & STORED ON MAGNETIC TAPE. PROGRAM IS RUN TWICE FOR EACH
C  FLIGHT ONCE TO OBTAIN ASCENT & DESCENT CAL VOLTAGES, ONCE TO PRODUCE
C  FINAL DATA OUTPUT.  WRITTEN MARCH 1974 IN FORTRAN IV LEVEL G BY
C  T. KNECHT.
C  MODIFIED FOR USE WITH WALLOPS TAPES SEPT. 1974.
C  UPDATED BY R.K. ZIMMERMAN AND M.K. MCINERNEY.
C
C  INPUT PARAMETER CARDS:
C    1: TAPE FORMAT PARAMETERS (ALL INTEGER EXCEPT SAMRAT):
C       SAMRAT = SAMPLING RATE PER SECOND (TOTAL FOR ALL CHANNELS).
C       IWDNUM = NO. OF WORD IN DATA FRAME ASSIGNED TO PROBE.
C       NCHANL = NO. OF CHANNELS SAMPLED.
C       NSMREC = TOTAL NO. OF WORDS PER DATA RECORD.
C       NHDCAL = NO. OF HEADER AND CALIBRATION RECORDS.
C       NID    = NO. OF HEADER (IDENTIFICATION) RECORDS.
C       CLRECL = NO. OF WORDS PER CALIBRATION RECORD.
C    2: ILOOP CONTROL PARM; COL 1: 0 FOR CAL RUNS, 1 FOR FINAL DATA RUNS
C    3: ICARDS CONTROL PARM; COL 1: 0 FOR PRINT ONLY, 1 FOR PRINT & CARD
C          OUTPUT.
C    4: AVERAGING PERIOD IN MSEC FOR EACH DATA POINT; INTEGER, COLS 1-5
C          (NORMALLY = SPIN PERIOD).
C    5: NO. OF OUTPUT POINTS/SECOND; INTEGER FACTOR OF 100, COLS 1-3
C          (E.G. 2,5,10,20,ETC).
C    6: GMT LAUNCH TIME IN HOURS, MINUTES & SECONDS; COLS 1-7, (HHMM:SS)
C    7: PRINTOUT HEADING LINE FOR DATA ID; ANY CHARACTERS, COLS 1-80
C    ** FOLLOWING 3 CARDS USED FOR FINAL DATA RUNS. OMIT FOR CAL RUNS **
C       8: PREFLIGHT CAL VOLTAGE; REAL, COLS 1-5 (X.XXX)
C       9: ASCENT CAL VOLTS, COLS 1-5 (X.XXX), AND GMT COLS 11-17
C             (HHMM:SS).
C       10: DESCENT CAL VOLTS AND TIME (SAME FORMAT AS PRECEDING CARD).
```

```
C    11: NCAL: NO. OF VOLTAGE/CURRENT CALIBRATION CARDS TO FOLLOW;
C          INTEGER, COLS 1-2.
C    12: DECK OF PROBE CALIBRATION VOLTAGE (V) & CURRENT (CALCUR) PAIRS:
C          VOLTS COLS 1-5 (X.XXX); CURRENT COLS 11-18 (X.XXE-NN).
C    13: DECK OF TRAJECTORY DATA (TIME IN SECONDS AFTER LAUNCH, ALTITUDE
C          IN KILOMETERS) -- 10 SECOND INCREMENTS;
C             TIME COLS 1-5 (XXX.X); ALTITUDE COLS 10-20 (XXX.XXX);
C             END OF DECK INDICATED BY CARD WITH TIME=0.
C
C          *********************************************************
C          *   - CAUTION  -  CAUTION  -  CAUTION  -  CAUTION  -    *
C          *   THE FINAL TRAJECTORY DATA CARD SHOULD BE THE TIME    *
C          *   WHEN THE ROCKET REACHES ZERO ALTITUDE, I.E., THE     *
C          *   CRASH TIME WITH ALTITUDE EQUAL ZERO.  FAILURE TO     *
C          *   INCLUDE THIS CARD MAY CAUSE SUBROUTINE TRAJ TO       *
C          *   UNDERFLOW, RESULTING IN PROGRAM TERMINATION.         *
C          *********************************************************
C
C    14: NEXT FOUR CARDS CONTAIN TIMES IN GMT, IN COLS 1-7 (HHMM:SS).
C       A:  START TIME OF ASCENT CAL (FOR CAL RUNS) OR TIME OF FIRST
C           DATA POINT (FOR FINAL DATA RUNS).
C       B:  STOP TIME OF ASCENT CAL (FOR CAL RUNS) OR TIME OF LAST
C           DATA POINT (FOR FINAL DATA RUNS).
C         C:  START TIME OF DESCENT CAL (OMIT FOR FINAL DATA RUNS).
C         D:  STOP TIME OF DESCENT CAL (OMIT FOR FINAL DATA RUNS).
C
C  SUBROUTINES REQUIRED:  CARDS,WCALIB,TRAJ,XYFCN,FINAL,WFETCH,SYNCRO,
C                         GETIME,UTIME,CONVER.
C  STORAGE REQUIREMENTS: COMPILATION, 116K BYTES; EXECUTION 75K BYTES
C
      COMMON/FETCH/TIMST,NCALL,NBLOCK,LHRS,LMINS,LSECS,LPTIME,
     *ARRAY
      COMMON/HOURS/LCHRS1
      COMMON/TAPE/SAMRAT,IWDNUM,NCHANL,NSMREC,NHDCAL,NID,CLRECL
      COMMON/BASE/HRB10,HRB1
      COMMON/TRJ/ATRJ,BTRJ,CTRJ,ITCALL,ALT,T
      REAL T(50),ALT(50),VOLTS(5),LCHRS,LCHMNS,LCHSCS,LTMSEC,INHRS,
     *INMINS,INSECS,LHRS,LMINS,LSECS,XDAT(6),YDAT(6),V(25),CALCUR(25),
     *PLOT(26)
      INTEGER*4 DATAID(20),RTINDX,RTSTOP,SUM(5),CALNO,DATSIZ,CLRECL
C
C - - CAUTION - - CAUTION - - CAUTION - - CAUTION - - CAUTION - -
C
C     DIMENSION OF ARRAY "DATA" MUST BE CHANGED IF NUMBER OF
C     SAMPLES PER SECOND EXCEEDS 2000.
C
C     DIMENSION OF ARRAY "ARRAY" MUST BE CHANGED IF LENGTH OF
C     DATA RECORD EXCEEDS 2008 WORDS.
C
      INTEGER*2 DATA(2000),ARRAY(2008),HRB10,HRB1
      NAMELIST/FORM/SAMRAT,IWDNUM,NCHANL,NSMREC,NHDCAL,NID,CLRECL
C  INITIALIZE VARIABLES.
      DATA BLANK,POINT/1H ,1H*/
      CALNO = 0
```

```
      ITCALL=0
      NCALL = 0
      IHGT=0
      LPTIME=0
      NBLOCK = 0
      ALTUDE=0.0
      CVTVOL=0.0
      ICARDS=0
      CURENT=0.0
      ICD=1
C  INSERT BLANKS IN PLOTTING ARRAY.
      DO 10 J=1,26
         PLOT(J)=BLANK
   10 CONTINUE
C  READ TAPE FORMAT PARAMETERS.
      READ (5,FORM)
      WRITE(11,FORM)
C  READ CONTROL PARMS, AVERAGING PERIOD, NO. POINTS/SEC, & LAUNCH TIME.
      READ(5,1000) ILOOP,ICARDS,IPERID,NPS,ILCH,ILCM,ILCS
 1000 FORMAT (I1/I1/I5/I3/2I2,1X,I2)
C  COMPUTE TIME INCREMENT IN CENTISECONDS.
      ITINCR = 100/NPS
C  CHECK FOR ACCEPTABLE VALUE OF NPS.
      RNPSCK = 100./NPS
      RTINCR = FLOAT(ITINCR)
      IF(RNPSCK.EQ.RTINCR) GOTO 20
      WRITE (6,1100)
 1100 FORMAT (/1X,'NUMBER OF POINTS/SECOND MUST BE INTEGER FACTOR OF 100
     *'/)
      STOP 22
C  CONVERT LAUNCH TIME TO REALS.
   20 LCHRS=FLOAT(ILCH)
      LCHRS1=LCHRS
      LCHMNS=FLOAT(ILCM)
      LCHSCS=FLOAT(ILCS)
      ILCGMT=100*ILCH + ILCM
C  INPUT AND PRINT DATA IDENTIFICATION HEADING.
      READ(5,1200) DATAID
 1200 FORMAT (20A4)
      WRITE (6,1300) DATAID
 1300 FORMAT (1H1,4X,20A4//)
C  IDENTIFY OUTPUT AS CAL OR FINAL DATA RUN.
      IF(ILOOP.EQ.1) GOTO 30
      WRITE (6,1400)
 1400 FORMAT (1X,'CALIBRATION RUN'//)
      GOTO 40
   30 WRITE (6,1500)
 1500 FORMAT (1X,'FINAL DATA RUN'//)
C  PRINT AVERAGING PERIOD, NUMBER OF POINTS/SECOND, AND LAUNCH
C  TIME (IF SECONDS HAS ONE DIGIT).
   40 IF(ILCS.GE.10) GOTO 50
      WRITE (6,1600) IPERID,NPS,ILCGMT,ILCS
 1600 FORMAT(1X,'AVERAGING PERIOD:  ',I3,' MSEC.'//1X,'NUMBER OF POINTS
     *PER SECOND:  ',I3//1X,'LAUNCH TIME:  ',I4,':0',I1)
```

```
      GOTO 60
C  PRINT AVERAGING PERIOD, NUMBER OF POINTS/SECOND, AND LAUNCH
C  TIME (IF SECONDS HAS TWO DIGITS).
   50 WRITE (6,1700) IPERID,NPS,ILCGMT,ILCS
 1700 FORMAT(1X,'AVERAGING PERIOD:  ',I3,' MSEC.'//1X,'NUMBER OF POINTS
     *PER SECOND:  ',I3//1X,'LAUNCH TIME:  ',I4,':',I2)
C  CHECK FOR ACCEPTABLE VALUE OF AVERAGING PERIOD (IPERID).
   60 IPERCK = (6000/NPS)-2
      IF(NPS.EQ.1.AND.IPERID.GT.1998) GOTO 70
      IF((NPS.EQ.2.OR.NPS.EQ.4).AND.IPERID.GT.998) GOTO 70
      IF(NPS.GE.5.AND.IPERID.GT.IPERCK) GOTO 70
      GOTO 80
   70 WRITE (6,1800)
 1800 FORMAT (1X,'INPUT ERROR--AVERAGING PERIOD MUST NOT BE GREATER THAN
     * VALUES SHOWN IN FOLLOWING TABLE'//10X,'POINTS/SECOND',7X,'PERIOD'
     *//16X,'1',14X,'1998'/16X,'2',15X,'998'/16X,'4',15X,'998'/13X,'5 TO
     * 100',6X,'(6000/NPS)-2'/)
      STOP 1
C  COMPUTE INTERVAL BETWEEN DATA SAMPLES IN MILLISECONDS.
   80 SAMINT = (NCHANL/SAMRAT) * 1000.
C  COMPUTE NO. OF DATA SAMPLES PER HALF SPIN PERIOD.
      IHPRD=IPERID/2
      ISAMP = IHPRD/SAMINT
      RLDIV = FLOAT(2*ISAMP+1)
C  CHECK THAT AVERAGING PERIOD IS AN EVEN INTEGER.
      IDHPRD=2*IHPRD
      IF(IDHPRD.EQ.IPERID) GOTO 90
      WRITE (6,1900) IDHPRD
 1900 FORMAT (1X,'PERIOD SUPPLIED NOT DIVISIBLE BY 2. PERIOD USED= ',
     *I4)
C  IF FINAL DATA RUN, READ PREFLIGHT, ASCENT, & DESCENT CAL VALUES.
   90 IF(ILOOP.EQ.0) GOTO 100
      CALL FINAL (PREFLV,UPCLV,UPTIM,DNCLV,DNTIM)
C  CALCULATE DRIFT CORRECTION COEFFICIENTS.
      A= (DNCLV-UPCLV)/(DNTIM-UPTIM)
      B=(PREFLV-UPCLV)
C  READ & PRINT VOLTAGE/CURRENT CALIBRATION PAIRS, FIND LOG OF CURRENTS.
  100 WRITE (6,2000)
 2000 FORMAT (/1X,'PROBE VOLTAGE/CURRENT CALIBRATION'//5X,'VOLTAGE',
     *9X,'CURRENT'/)
      READ (5,2100) NCAL
 2100 FORMAT (I2)
      DO 110 IKL=1,NCAL
         READ (5,2200) V(IKL),CALCUR(IKL)
         WRITE(6,2300) V(IKL),CALCUR(IKL)
         CALCUR(IKL)=ALOG(CALCUR(IKL))
  110 CONTINUE
 2200 FORMAT (F5.3,5X,E8.2)
 2300 FORMAT(6X,F5.3,9X,1PE8.2)
C  OPEN TAPE FILE.
      CALL TPOPIZ(12)
C  READ, COMPUTE & PRINT TM SYSTEM STEP CALIBRATIONS.
      CALL WCALIB(XDAT)
      XDAT(6) = 2*XDAT(5) - XDAT(4)
```

```
      WRITE (6,2400) XDAT
 2400 FORMAT(//6X,'TM SYSTEM CALIBRATION'//5X,'LBE (-7.50%)',4X,F7.1//5X
     *,'LHB (-3.75%)',4X,F7.1//5X,'BAND CENTER',5X,F7.1//5X,'UHB (+3.75%
     *)',4X,F7.1//5X,'UBE (+7.50%)',4X,F7.1//5X,'EXTRAP. PT.',5X,F7.1)
C  READ & PRINT TRAJECTORY DATA.
      WRITE (6,1300) DATAID
      WRITE (6,2500)
 2500 FORMAT(/9X,'TRAJECTORY DATA'//7X,'TIME',10X,'ALTITUDE'/)
  120 IHGT=IHGT+1
      READ (5,2600) T(IHGT),ALT(IHGT)
 2600 FORMAT (F9.1,F11.3)
      WRITE (6,2700) T(IHGT),ALT(IHGT)
 2700 FORMAT(6X,F5.1,10X,F7.3)
      IF(T(IHGT).GT.0) GOTO 120
C  INITIALIZE TM VOLTAGE CONVERSION ARRAY.
      YDAT(1)=0.0
      YDAT(2)=1.25
      YDAT(3)=2.50
      YDAT(4)=3.75
      YDAT(5)=5.00
      YDAT(6)=6.25
C  PRINT HEADINGS.
      WRITE (6,1300) DATAID
      WRITE (6,2800)
 2800 FORMAT(8X,'TIME',7X,'SECONDS',4X,'ALTITUDE',5X,'FINAL',6X,'AVERAGE
     *',2X,'AVERAGE',3X,'CORRECTED',7X,'CORRECTED  VOLTAGE',10X,'RECORD'
     */8X,'(GMT)',4X,'FROM LAUNCH',4X,'(KM)',6X,'CURRENT',5X,'A/D NO.',2
     *X,'VOLTAGE',4X,'VOLTAGE',4X,'0----1----2----3----4----5',6X,'NUMBE
     *R'/)
C  CONVERT LAUNCH TIME TO SECONDS.
      LCHRS=0.
      CALL CONVER (LCHRS,LCHMNS,LCHSCS,LTMSEC)
C  READ START & STOP TIMES AND CONVERT TO SECONDS.
  130 READ (5,2900) ISTH,ISTM,ISTS,IENDH,IENDM,IENDS
 2900 FORMAT (2I2,1X,I2)
      IF(CALNO.EQ.1) GOTO 140
      HRBASE=ISTH
      HRB10=HRBASE/10
      HRB1=HRBASE-HRB10*10
      HRB10=HRB10*16
  140 ISTH=ISTH-HRBASE
      STHRS=FLOAT(ISTH)
      STMINS=FLOAT(ISTM)
      STSECS=FLOAT(ISTS)
      CALL CONVER (STHRS,STMINS,STSECS,TIMSEC)
      IENDH=IENDH-HRBASE
      ENDHRS=FLOAT(IENDH)
      ENDMNS=FLOAT(IENDM)
      ENDSCS=FLOAT(IENDS)
      CALL CONVER(ENDHRS,ENDMNS,ENDSCS,ENDTIM)
C  COMPUTE TIME OF FIRST DATA BLOCK REQUIRED.
      TIMST=TIMSEC-1.0
C  ESTABLISH INTEGER TO KEEP TRACK OF TIME FOR EACH DATA POINT.
      ITMSEC=100*IFIX(TIMSEC)
```

```
C  CALL FIRST ONE-SECOND ARRAY OF DATA SAMPLES.
      CALL WFETCH(DATSIZ,DATA)
C  FIND TIME INCREMENT BETWEEN DATA POINTS.
      INCR=1000/NPS
      SINCR = INCR / SAMINT
      ISI = SINCR
      DIFR = SINCR - ISI
      IF(DIFR.GT.0.5) ISI = ISI + 1
      RINCR=FLOAT(INCR)
      INDEX = DATSIZ
  150 IRUN=1
      NSUM=1
      NUMSUM=0
      SUM(1)=0
      SUM(2)=0
      SUM(3)=0
      SUM(4)=0
      SUM(5)=0
C  COMPUTE SUBSCRIPTS OF FIRST & LAST SAMPLES FOR AVERAGING PERIOD.
      LFINDX = INDEX - ISAMP
      RTINDX = INDEX + ISAMP
C  IF ALL SAMPLES WITHIN AVERAGING PERIOD ARE IN PRESENT ARRAY, SUM THEM.
C  IF SAMPLES OVERLAP NEXT ARRAY, SUM THOSE IN PRESENT ARRAY, THEN CALL
C  NEW ARRAY & COMPLETE SUM. REPEAT FOR ALL DATA POINTS WHOSE AVERAGING
C  PERIODS SPAN THE TIME BOUNDARY BETWEEN TWO SUCCESSIVE ARRAYS.
      IF(RTINDX.GT.DATSIZ) GOTO 160
      ISTOP=RTINDX
      NRTA=1
      GOTO 170
  160 ISTOP=DATSIZ
      NRTA=0
  170 NUMSUM=NUMSUM+1
      DO 180 I=LFINDX,ISTOP
         SUM(NUMSUM) = SUM(NUMSUM) + DATA(I)
  180 CONTINUE
      IF(NRTA.GT.0) GOTO 220
      LFINDX = LFINDX + ISI
      IF(LFINDX.GE.(DATSIZ+1)) GOTO 190
      GOTO 170
  190 RTINDX = RTINDX - DATSIZ
      CALL WFETCH(DATSIZ,DATA)
  200 DO 210 I=1,RTINDX
         SUM(NSUM)=SUM(NSUM)+DATA(I)
  210 CONTINUE
C  CONVERT SUMS TO REALS AND FIND AVERAGE. STORE IN ARRAY 'VOLTS'.
  220 RLSUM=FLOAT(SUM(NSUM))
      VOLTS(NSUM)=RLSUM/RLDIV
      IF(NSUM.EQ.NUMSUM) GOTO 230
      RTINDX = RTINDX + ISI
      NSUM = NSUM + 1
      GOTO 200
C  CONVERT AVERAGE TM OUTPUT VOLTAGE TO EQUIV. VCO INPUT VOLTAGE (0-5 V)
  230 PRBVOL = VOLTS(1)
  240 IWARN = 0
```

```
      CALL XYFCN (XDAT,YDAT,6,PRBVOL,IWARN,CVTVOL)
C  IF FINAL DATA RUN, APPLY DRIFT CORRECTION DERIVED FROM INFLIGHT CALS.
      IF(ILOOP.EQ.1) GOTO 250
      DELTA = 0.
      GOTO 260
  250 DELTA= -A*(TIMSEC-UPTIM)+B
  260 CORVLT = CVTVOL + DELTA
C  CONVERT PROBE OUTPUT VOLTAGE TO CORRESPONDING LOG CURRENT VALUE.
      CALL XYFCN (V,CALCUR,NCAL,CORVLT,IWARN,CURENT)
C  CONVERT LOG CURRENT TO CURRENT.
      YCURNT = EXP(CURENT)
C  FIND GMT FOR PRESENT DATA POINT.  CONVERT HOURS & MINS TO INTEGERS.
C  FOR PRINTING, COMPUTE TIME IN SECONDS AFTER LAUNCH (SECALC).
      CALL UTIME (TIMSEC,UTIMEH,UTIMEM,UTIMES)
      SECALC = TIMSEC - LTMSEC
      IUTIMH=IFIX(UTIMEH)
      IUTIMM=IFIX(UTIMEM)
      ITMGMT=(HRBASE+IUTIMH)*100  +  IUTIMM
C  POSITION PLOTTING POINT.
      K=(CORVLT*5.0)+1.0
      IF(K.GT.26) K=26
      IF(K.LT.1) K=1
      PLOT(K)=POINT
C  COMPUTE ALTITUDE FOR PRESENT OUTPUT POINT.
      CALL TRAJ (SECALC,ALTUDE)
C  SELECT OUTPUT FORMAT AND PRINT RESULTS.
      IF(UTIMES.GE.10.) GOTO 280
      IF(IWARN.EQ.1) GOTO 270
      WRITE(6,3000) ITMGMT,UTIMES,SECALC,ALTUDE,YCURNT,PRBVOL,CVTV
     *OL,CORVLT,PLOT,NBLOCK
 3000 FORMAT(6X,I4,':0',F4.2,4X,F6.2,4X,F7.3,4X,1PE9.3,4X,0PF7.1,3X,F6.4
     *,5X,F6.4,3X,'I',26A1,'I',6X,I4)
      GOTO 300
  270 WRITE (6,3100) ITMGMT,UTIMES,SECALC,ALTUDE,NBLOCK
 3100 FORMAT(6X,I4,':0',F4.2,4X,F6.2,4X,F7.3,4X,'* VOLTAGE NOT WITHIN CA
     *LIBRATION RANGE *',3X,'I',26X,'I',6X,I4)
      GOTO 300
  280 IF(IWARN.EQ.1) GOTO 290
      WRITE(6,3200) ITMGMT,UTIMES,SECALC,ALTUDE,YCURNT,PRBVOL,CVTV
     *OL,CORVLT,PLOT,NBLOCK
 3200 FORMAT(6X,I4,':',F5.2,4X,F6.2,4X,F7.3,4X,1PE9.3,4X,0PF7.1,3X,F6.4
     *,5X,F6.4,3X,'I',26A1,'I',6X,I4)
      GOTO 300
  290 WRITE (6,3300) ITMGMT,UTIMES,SECALC,ALTUDE,NBLOCK
 3300 FORMAT (6X,I4,':',F5.2,4X,F6.2,4X,F7.3,4X,'* VOLTAGE NOT WITHIN CA
     *LIBRATION RANGE *',3X,'I',26X,'I',6X,I4)
C  IF CARD OUTPUT REQUIRED, TRANSFER OUTPUT TO PUNCHING SUBROUTINE.
  300 IF(ICARDS.EQ.1) CALL CARDS(ICD,SECALC,YCURNT)
C  REZERO PLOT ARRAY.
      PLOT(K)=BLANK
C  INCREMENT TIME FOR NEXT DATA POINT. TRUNCATE TO FIND SECONDS, 10THS &
C  100THS OF SECONDS.
      IRUN=IRUN+1
      ITMSEC=ITMSEC+ITINCR
```

```
      TIMSEC=FLOAT(ITMSEC)/100.
      ITMTRN=(ITMSEC/10)*10
      ITM100=ITMSEC-ITMTRN
      ITMTR2=(ITMSEC/100)*100
      ITM10=(ITMSEC-ITMTR2)/10
      ITMTR3=(ITMSEC/1000)*1000
      ITM1=(ITMSEC-ITMTR3)/100
C  SPACE BETWEEN GROUPS OF DATA POINTS. GROUP SIZE DETERMINED BY NPS.
      IF(NPS.LE.4.AND.ITM1.EQ.0) WRITE (6,3400)
      IF(NPS.GE.5.AND.NPS.LE.25.AND.ITM10.EQ.0) WRITE (6,3400)
      IF(NPS.GE.50.AND.ITM100.EQ.0) WRITE (6,3400)
 3400 FORMAT  (84X,'I',26X,'I')
C  CHECK THAT TIME OF ARRAY CORRESPONDS TO TIME OF PRESENT DATA POINT.
C  OMIT CHECK FOR POINTS WHOSE SAMPLES OVERLAP DATA ARRAYS.
      IF(ITM10.LT.2.OR.ITM10.GT.8) GOTO 310
      CALL CONVER (LHRS,LMINS,LSECS,TIMCHK)
      ITMCHK=100*IFIX(TIMCHK)
      IF(ITMCHK.EQ.ITMTR2) GOTO 310
      CALL UTIME (TIMSEC,UTIMEH,UTIMEM,UTIMES)
      ILHRS=IFIX(LHRS)
      ILMINS=IFIX(LMINS)
      IUTIMH=IFIX(UTIMEH)
      IUTIMM=IFIX(UTIMEM)
      WRITE(6,3500) ILHRS,ILMINS,LSECS,IUTIMH,IUTIMM,UTIMES
 3500 FORMAT (1X,'***TIMING ERROR***'/1X'TIME OF PRESENT FETCH BLOCK: ',
     *2I2,':',F5.2,/1X,'TIME OF PRESENT DATA POINT: ',2I2,':',F5.2)
C  CHECK FOR TIME OF LAST DATA POINT.
  310 IF(TIMSEC.GT.ENDTIM) GOTO 330
C  IF DATA POINT OVERLAPS TWO ARRAYS, COMPUTE CURRENT FOR NEXT AVERAGE.
      IF(IRUN.GT.NUMSUM) GOTO 320
      PRBVOL=VOLTS(IRUN)
      GOTO 240
C  DETERMINE SUBSCRIPTS OF FIRST & LAST SAMPLES FOR NEXT DATA POINT.
  320 ITINDX = (10*ITM100 + 100*ITM10) / SAMINT
      IF(ITINDX.EQ.0) ITINDX=DATSIZ
      INDEX=ITINDX
      GOTO 150
C  IF FINAL DATA RUN, STOP. IF CAL RUN, READ START & STOP TIMES FOR
C  DESCENT CAL, SKIP TO TOP OF PAGE & RE-ENTER PROCESSING ROUTINE.
  330 IF(ILOOP.EQ.1) GOTO 340
      CALNO=CALNO+1
C  STOP AFTER DESCENT CAL.
      IF(CALNO.GT.1) GOTO 350
      NCALL=0
      WRITE(6,3600)
 3600 FORMAT(1H1)
      WRITE(6,2800)
      GOTO 130
  340 CALL TPCLSZ(12)
      STOP 3
  350 CALL TPCLSZ(12)
      STOP 4
      END
      SUBROUTINE CARDS (ICD,SECALC,YCURNT)
```

```
C
C  PUNCH 1 TIME AND 5 CURRENTS PER CARD
C  TIME IS OF FIRST CURRENT, FOLLOWING CURRENTS ARE AT 0.1 SEC INTERVALS
C   INPUT ARGUMENTS:    SECALC = TIME IN SECONDS AFTER LAUNCH
C                       YCURNT = PROBE CURRENT
C
      REAL STORE(10)
C PLACE TIME/CURRENT PAIR INTO STORAGE ARRAY
      STORE(ICD)=SECALC
      STORE(ICD+1) = YCURNT
      ICD = ICD + 2
C  IF ARRAY NOT FULL,RETURN TO MAIN PROGRAM FOR NEW DATA.  IF FULL,
C  PUNCH DATA CARD, THEN RETURN.
      IF(ICD.LT.11) RETURN
      WRITE(7,1000) STORE(1),(STORE(KK),KK=2,10,2)
 1000 FORMAT(F7.3,5E13.4)
      ICD = 1
      RETURN
      END
      SUBROUTINE WCALIB(XDAT)
C
C  THIS SUBROUTINE REPLACES THE CALIBR SUBROUTINE USED IN THE ORIGINAL
C  PROBE PROGRAM.  IT ALLOWS THE PROGRAM TO PROCESS DATA FROM DIGITAL
C  TAPES PREPARED AT THE WALLOPS ISLAND PCM FACILITY.  IT ASSUMES THAT
C  THE FIVE CALIBRATION RECORDS IMMEDIATELY FOLLOW THE IDENTIFICATION
C  HEADER RECORDS.
C
C  NID IS THE NUMBER OF IDENTIFICATION RECORDS.
C  CLRECL IS THE NUMBER OF WORDS PER CALIBRATION RECORD.
C
      COMMON/FETCH/TIMST,NCALL,NBLOCK,LHRS,LMINS,LSECS,LPTIME,
     *ARRAY
      COMMON/TAPE/SAMRAT,IWDNUM,NCHANL,NSMREC,NHDCAL,NID,CLRECL
      REAL XDAT(6)
      INTEGER*4 CLRECL
      INTEGER*2 ARRAY(1)
      ASSIGN 40 TO KILL
C  READ AND PRINT FIRST IDENTIFICATION HEADER RECORD.
      CALL TPGETZ(12,ARRAY(1))
      CALL TPCHKZ(12,NTAPE,KILL)
      NBLOCK = NBLOCK + 1
      WRITE(6,1000) (ARRAY(M),M=1,45)
 1000 FORMAT(//,1X,'TAPE INFORMATION: ',45A2)
C  SKIP REMAINING THREE HEADER RECORDS.
      JP=NID-1
      DO 10 J=1,JP
         CALL TPFSRZ(12)
         NBLOCK = NBLOCK + 1
   10 CONTINUE
C  IDX IS THE SUBSCRIPT OF THE FIRST CALIBRATION SAMPLE IN THE RECORD.
      IDX = IWDNUM + 5
C  READ CAL RECORDS AND FIND AVERAGE FOR EACH ONE.
      DO 30 ISTEP = 1,5
         CALL TPGETZ(12,ARRAY(1))
```

```
         CALL TPCHKZ(12,NTAPE,KILL)
         NBLOCK = NBLOCK + 1
         ISUM = 0
         DO 20 L = IDX,CLRECL,NCHANL
            ISUM = ISUM + ARRAY(L)
   20    CONTINUE
         RSUM = ISUM
         XDAT(ISTEP) = RSUM*NCHANL/(CLRECL-5)
   30 CONTINUE
      RETURN
   40 WRITE(6,1100) NTAPE
 1100 FORMAT(' ',' TAPE ERROR ', I8)
      STOP
      END
      SUBROUTINE TRAJ (TIME,HEIGHT)
C
C  COMPUTES INTERPOLATED ALTITUDE VALUE FOR ANY TIME AFTER LAUNCH.
C  MODIFIED FOR USE WITH PROGRAM PROBE.  ALTITUDE AND TIME VALUES AT
C  10-SECOND INCREMENTS MUST BE PROVIDED VIA ARRAYS T (TIME) AND ALT
C  (ALTITUDE) FOR ENTIRE PERIOD OF FLIGHT.
C  INPUT ARGUMENT:  TIME = TIME IN SECONDS AFTER LAUNCH.
C  OUTPUT ARGUMENT: HEIGHT = ALTITUDE IN KILOMETERS.
C
      COMMON/TRJ/A,B,C,ITCALL,ALT,T
      REAL ALT(50),T(50)
      INCR=0
C  IF FIRST CALL TO TRAJ, INITIALIZE VARIABLES.  IF NOT,SKIP TO 20.
      IF(ITCALL.GT.0) GOTO 20
      I=3
      ITCALL=1
   10 INCR=1
      IM1=I-1
      IM2=I-2
   20 IF(TIME.LT.T(IM2)) WRITE (6,1000)
 1000 FORMAT(/1X,'TIME LESS THAN LOWEST TRAJECTORY POINT')
C  LOCATE TIME VALUES WHICH BRACKET PRESENT INPUT TIME VALUE.
      IF(TIME.LE.T(I)) GOTO 30
      I=I+1
      GOTO 10
C  IF PRESENT TIME VALUE IS IN SAME INTERVAL AS PREVIOUS ONE, COMPUTE
C  ALTITUDE WITH OLD COEFFICIENTS. IF NOT, COMPUTE NEW COEFF'S FIRST.
   30 IF(INCR.EQ.1) GOTO 50
   40 HEIGHT=A*TIME*TIME+B*TIME+C
      RETURN
   50 BRAC1=(T(I)-T(IM1))*(ALT(IM1)-ALT(IM2))
      BRAC2=(T(IM1)-T(IM2))*(ALT(I)-ALT(IM1))
      TOP=BRAC1-BRAC2
      BRAC1=(T(IM1)-T(IM2))*(T(I)*T(I)-T(IM1)*T(IM1))
      BRAC2=(T(I)-T(IM1))*(T(IM1)*T(IM1)-T(IM2)*T(IM2))
      BOTTOM=BRAC2-BRAC1
      A=TOP/BOTTOM
      B=(ALT(IM1)-ALT(IM2))-A*(T(IM1)*T(IM1)-T(IM2)*T(IM2))
      B=B/(T(IM1)-T(IM2))
      C=ALT(IM2)-A*T(IM2)*T(IM2)-B*T(IM2)
```

```
      GOTO 40
      END
      SUBROUTINE XYFCN (XDAT,YDAT,NDATA,X,IWARN,Y)
C
C  LINEAR INTERPOLATION ROUTINE.  DEVELOPED BY GCA, MODIFIED FOR U OF I
C  USE WITH PROGRAM PROBE.
C  INPUT ARGUMENTS:  XDAT, YDAT = PARALLEL ARRAYS OF X & Y COORD. PAIRS.
C                    NDATA = NO. OF COORDINATE PAIRS SUPPLIED.
C                    X = VALUE OF X FOR WHICH Y IS TO BE FOUND.
C  OUTPUT ARGUMENTS: IWARN = FLAG SET TO 1 IF INPUT VALUE OF X EXCEEDS
C                            RANGE OF SUPPLIED X COORDINATES.
C                    Y = INTERPOLATED VALUE CORRESPONDING TO INPUT X.
C
      REAL XDAT(25),YDAT(25)
C  DETERMINE IF INPUT X IS <, =, OR > LOWEST XDAT SUPPLIED.
      IF(X-XDAT(1))40,10,20
   10 Y=YDAT(1)
      GOTO 80
C  DETERMINE IF INPUT X IS <, =, OR > HIGHEST XDAT SUPPLIED.
   20 IF(X-XDAT(NDATA)) 50,30,40
   30 Y=YDAT(NDATA)
      GOTO 80
C  IF INPUT X NOT WITHIN RANGE OF SUPPLIED VALUES, SET WARNING FLAG.
   40 IWARN = 1
      GOTO 80
C  LOCATE 2 XDAT VALUES WHICH BRACKET THE INPUT X VALUE.
C  COMPUTE INTERPOLATED VALUE FOR Y.
   50 DO 60 I=2,NDATA
         IM=I-1
         IF(X-XDAT(I)) 70,60,60
   60 CONTINUE
   70 Y=YDAT(IM)+(YDAT(I)-YDAT(IM))*(X-XDAT(IM))/(XDAT(I)-XDAT(IM))
   80 RETURN
      END
      SUBROUTINE FINAL (PREFLV,UPCLV,UPTIM,DNCLV,DNTIM)
C
C  INPUTS PREFLIGHT AND INFLIGHT CALIBRATION DATA DURING FINAL DATA RUNS
C  OF PROGRAM PROBE.
C  OUTPUT ARGUMENTS:  PREFLV = PREFLIGHT CALIBRATION VOLTAGE.
C                     UPCLV = ASCENT CALIBRATION VOLTAGE.
C                     UPTIM = TIME OF ASCENT CALIBRATION.
C                     DNCLV = DESCENT CALIBRATION VOLTAGE.
C                     DNTIM = TIME OF DESCENT CALIBRATION.
C
C  READ AND PRINT PREFLIGHT CALIBRATION VOLTAGE.
      READ (5,1000) PREFLV
 1000 FORMAT (F5.3)
      WRITE(6,1100) PREFLV
 1100 FORMAT(/1X,'PREFLIGHT CALIBRATION VOLTAGE:  ',F7.3)
C  READ AND PRINT ASCENT CALIBRATION VOLTAGE AND TIME.
      COMMON/HOURS/LCHRS1
      READ(5,1200) UPCLV,ITIMUH,ITIMUM,ITIMUS
 1200 FORMAT (F5.3,5X,2I2,1X,I2)
      WRITE(6,1300) UPCLV,ITIMUH,ITIMUM,ITIMUS
```

```
 1300 FORMAT(/1X,'ASCENT CALIBRATION VOLTAGE:  ',3X,F7.3,10X,'TIME:  ',2
     *I2,':',I2)
C  CONVERT TIME TO SECONDS.
      TIMEUH=FLOAT(ITIMUH)-LCHRS1
      TIMEUM=FLOAT(ITIMUM)
      TIMEUS=FLOAT(ITIMUS)
      CALL CONVER (TIMEUH,TIMEUM,TIMEUS,UPTIM)
C  READ AND PRINT DESCENT CALIBRATION VOLTAGE AND TIME.
      READ (5,1200) DNCLV,ITIMDH,ITIMDM,ITIMDS
      WRITE (6,1400) DNCLV,ITIMDH,ITIMDM,ITIMDS
 1400 FORMAT (/1X,'DESCENT CALIBRATION VOLTAGE:  ',2X,F7.3,10X,'TIME:
     *,2I2,':',I2)
C  CONVERT TIME TO SECONDS.
      TIMEDH=FLOAT(ITIMDH)-LCHRS1
      TIMEDM=FLOAT(ITIMDM)
      TIMEDS=FLOAT(ITIMDS)
      CALL CONVER (TIMEDH,TIMEDM,TIMEDS,DNTIM)
      RETURN
      END
      SUBROUTINE WFETCH (DATSIZ,DATA)
C
C  THIS SUBROUTINE REPLACES THE FETCH SUBROUTINE USED IN THE ORIGINAL
C  PROBE PROGRAM.  IT ALLOWS THE PROGRAM TO PROCESS DATA FROM DIGITAL
C  TAPES PREPARED AT THE WALLOPS ISLAND PCM FACILITY.  IT ASSUMES THAT
C  EACH DATA RECORD STARTS WITH FIVE ID WORDS AND ENDS WITH THREE TIME
C  WORDS.
C
C  IWDNUM IS THE NUMBER OF THE WORD IN THE DATA FRAME ASSIGNED TO THE
C  PROBE SAMPLES.
C  SAMRAT IS THE OVERALL SAMPLING RATE (NO. OF SAMPLES/CHANNEL/SECOND
C  TIMES NO. OF CHANNELS)
C  NHDCAL IS THE TOTAL NUMBER OF HEADER AND CALIBRATION RECORDS.
C  NCHANL IS THE NUMBER OF DATA CHANNELS RECORDED (NO. OF WORDS PER DATA
C  FRAME).
C  NSMREC IS THE NUMBER OF WORDS PER DATA RECORD (INCL. FIVE ID AND
C  THREE TIME WORDS).
C  CLRECL IS THE NUMBER OF WORDS IN EACH CALIBRATION RECORD.
C
      COMMON/FETCH/TIMST,NCALL,NBLOCK,LHRS,LMINS,LSECS,LPTIME,
     *ARRAY
      COMMON/AREA1/OLDTIM,MEAN,IIFLAG
      COMMON/TAPE/SAMRAT,IWDNUM,NCHANL,NSMREC,NHDCAL,NID,CLRECL
      COMMON/BASE/HRB10,HRB1
      REAL LHRS,LMINS,LSECS,LENGTH
      INTEGER*4 HRS,RECHRS,RECMIN,DLAST,DATSIZ,CLRECL
      INTEGER*2 ARRAY(1),DATA(1),HRB10,HRB1
      DOUBLE PRECISION CVTTIM,SEARCH,RECTIM,ENDREC,BLKTIM,OLDTIM,MEAN
      CVTTIM (HRS,MINS,SECS) = 3.6D3*HRS + 6.D1*MINS + 1.D0*SECS
      ASSIGN 110 TO KILL
      IF(NCALL.GT.0) GOTO 50
      SEARCH = TIMST
C  COMPUTE LENGTH OF DATA RECORD IN SECONDS AND SUBSCRIPT OF LAST DATA
C  SAMPLE.
      LENGTH = (NSMREC - 8)/SAMRAT
```

```
      DLAST = NSMREC - 3
C  MAKE SURE THAT ALL HEADER AND CALIBRATION RECORDS HAVE BEEN READ.
      IF(NBLOCK.GE.NHDCAL) GOTO 20
      IJMN=NHDCAL-NBLOCK
      DO 10 LMN=1,IJMN
         CALL TPFSRZ(12)
   10 CONTINUE
      NBLOCK=NBLOCK+IJMN
      WRITE(6,1000) NBLOCK
 1000 FORMAT(1X,'RECORDS HAVE BEEN SKIPPED TO NBLOCK = ',I3)
C  IF FIRST TIME THROUGH, COMPUTE START TIME, READ DATA RECORD.
   20 ISRCH = SEARCH
   30 CALL TPGETZ(12,ARRAY(1))
      CALL TPCHKZ(12,NTAPE,KILL)
      NBLOCK = NBLOCK + 1
      CALL GETIME (RECHRS,RECMIN,RECSEC)
      RECHRS=RECHRS-10*(HRB10/16)-HRB1
      RECTIM = CVTTIM (RECHRS,RECMIN,RECSEC)
      CALL SYNCRO(RECTIM)
      ENDREC = RECTIM + LENGTH
      WRITE (11,1100) RECHRS,RECMIN,RECSEC,RECTIM,SEARCH,ENDREC
 1100 FORMAT(1X,'RECHRS= ',I2,3X,'RECMIN= ',I2,3X,'RECSEC= ',F7.4,3X,
     *   'RECTIM= ',F10.4,3X,'SEARCH= ',F10.4,3X,'ENDREC= ',F10.4)
C  IF START TIME IS EARLIER THAN FIRST RECORD, PRINT ERROR MESSAGE.
      IF(SEARCH.GE.RECTIM) GOTO 40
      WRITE (6,1200)
 1200 FORMAT (1X,'START TIME IS EARLIER THAN TIME OF FIRST RECORD.')
      STOP 7
   40 IF(SEARCH.GT.ENDREC) GOTO 30
      NCALL = 2
C  COMPUTE SUBSCRIPT OF FIRST SAMPLE REQUIRED.
      DIFFER = SEARCH - RECTIM
      INDEX = SAMRAT * DIFFER + NCHANL/2
      IDXFST = (INDEX/NCHANL) * NCHANL + 5 + IWDNUM
      WRITE (11,1300) RECTIM,ENDREC,SEARCH,NBLOCK,DIFFER
 1300 FORMAT(/1X,'RECTIM= ',F10.4,3X,'ENDREC= ',F10.4,3X,'SEARCH= ',
     *   F10.4,3X,'NBLOCK= ',I4,3X,'DIFFER= ',F10.4)
      WRITE (11,1400) INDEX,IDXFST,ISRCH
 1400 FORMAT (1X,'INDEX= ',I4,3X,'IDXFST= ',I4,3X,'ISRCH= ',I7/)
   50 I = 1
      BLKEND = ISRCH + 1.0
      ILAST = NSMREC
   60 BLKTIM = RECTIM + (IDXFST-3)/SAMRAT + .001
   70 IF(IDXFST.LE.DLAST) GOTO 90
      CALL TPGETZ(12,ARRAY(1))
      CALL TPCHKZ(12,NTAPE,KILL)
      IDXFST = IWDNUM + 5
      NBLOCK = NBLOCK + 1
      CALL GETIME (RECHRS,RECMIN,RECSEC)
      RECHRS=RECHRS-10*(HRB10/16)-HRB1
      RECTIM = CVTTIM(RECHRS,RECMIN,RECSEC)
      CALL SYNCRO(RECTIM)
      ENDREC = RECTIM + LENGTH
      IF(BLKEND.GT.ENDREC) GOTO 80
```

```
      DIFFER = BLKEND - RECTIM
      INDEX = SAMRAT * DIFFER + NCHANL/2
      ISUB = (INDEX/NCHANL) * NCHANL + 5 + IWDNUM
      ILAST = ISUB - NCHANL
      WRITE(11,1500) BLKEND,INDEX,ISUB,ILAST
 1500 FORMAT (1X,'BLKEND= ',F10.4,4X,'INDEX= ',I5,4X,'ISUB= ',
     *I5,4X,'ILAST= ',I5)
      WRITE(11,1600) ENDREC,RECTIM,IDXFST
 1600 FORMAT(1X,'ENDREC= ',F10.4,4X,'RECTIM= ',F10.4,4X,'IDXFST= ',
     *I5,4X)
   80 IF(I.EQ.1) GOTO 60
   90 IF(IDXFST.GT.ILAST) GOTO 100
      DATA(I) = ARRAY(IDXFST)
      I = I + 1
      IDXFST = IDXFST + NCHANL
      GOTO 70
  100 IDXFST = ISUB
      DATSIZ = I-1
      CALL UTIME (BLKTIM,LHRS,LMINS,RSECS)
      LSECS = AINT (RSECS)
C  INCREMENT SEARCH TIME FOR NEXT CALL TO WFETCH.
      ISRCH = ISRCH + 1
      RETURN
C  COME HERE IF TAPE ERROR IS DETECTED.
  110 WRITE(6,1700) NTAPE
 1700 FORMAT(' ',' TAPE ERROR  ', I8)
      STOP 9
      END
      SUBROUTINE SYNCRO(RECTIM)
C
C  SUBROUTINE SYNCRO IS DESIGNED TO BE USED WITH SUBROUTINE WFETCH. DATA
C  TAPES RECEIVED FROM WALLOPS ISLAND SOMETIMES HAVE SPORATIC TIME CODE
C  ERRORS. SINCE WFETCH READS THE TIME CODE OF EVERY RECORD, A SINGLE
C  BAD CODE HAD THE EFFECT OF STOPPING DATA PROCESSING.SUBROUTINE SYNCRO
C  WILL PERMIT DATA PROCESSING TO CONTINUE WITH WFETCH THROUGH PORTIONS
C  OF THE TAPE WITH BAD TIME CODES BY:
C     1.) IDENTIFYING BAD TIME CODES,
C     2.) INTERNALLY GENERATING THE PROPER TIME FOR A RECORD,
C     3.) SUBSTITUTING THE GENERATED TIME FOR USE IN WFETCH
C         UNTIL THE TAPE TIME CODES ARE ONCE AGAIN CORRECT.
C  WHENEVER SYNCRO SUBSTITUTES A GENERATED TIME FOR A TIME READ FROM A
C  TAPE RECORD, A NOTICE TO THAT EFFECT WILL APPEAR IN THE OUTPUT ALONG
C  WITH THE RECORD NUMBER.
C
C  SYNCRO WILL TERMINATE THE PROGRAM IF:
C     1.) A TIME CODE ERROR IS DETECTED IN THE FIRST 20 TAPE RECORDS,
C     2.) 50 OR MORE ERRORS ARE DETECTED.
C
C  UNIVERSITY OF ILLINOIS AERONOMY LABORATORY
C  R.K.ZIMMERMAN,JR.-AUTHOR. 9 SEPTEMBER 1976
C
      COMMON/FETCH/TIMST,NCALL,NBLOCK,LHRS,LMINS,LSECS,LPTIME,
     *ARRAY
      COMMON/AREA1/OLDTIM,MEAN,IIFLAG
```

```
      COMMON/TAPE/SAMRAT,IWDNUM,NCHANL,NSMREC,NHDCAL,NID,CLRECL
      INTEGER*4 CLRECL
      INTEGER*2 ARRAY(1)
      DOUBLE PRECISION RECTIM,OLDTIM,DELTA,MEAN
C  IS THIS THE FIRST CALL TO SYNCRO?
      IF(NBLOCK.EQ.(NHDCAL+1)) GOTO 30
      DELTA= RECTIM - OLDTIM
C  IS THE TIME DIFFERENCE THE EXPECTED LENGTH?
      IF(((1.8*MEAN).GT.DELTA).AND.((0.2*MEAN).LT.DELTA)) GOTO 40
      IF((DELTA.GT.0.0).AND.(DELTA.LT.(10.0*MEAN))) GOTO 60
      WRITE(11,1000) NBLOCK
 1000 FORMAT(' '/' ','LOSS OF SYNCHRONIZATION IN RECORD NUMBER ',I5,
     *'.   INTERNALLY GENERATED TIME SUBSTITUTED.')
C  DID THE FAILURE OCCUR IN THE FIRST 20 RECORDS?
      IF(NBLOCK.LT.20) GOTO 20
C  GENERATE NEW RECTIM.
      RECTIM= OLDTIM + MEAN
   10 IIFLAG= IIFLAG + 1
      IF(IIFLAG.LT.50) GOTO 50
      WRITE(6,1100)
 1100 FORMAT(' '/' ','LOSS OF SYNCHRONIZATION FOR A TOTAL OF 50 RECORDS.
     * AUTOMATIC PROGRAM TERMINATION')
      STOP
C  COME HERE IF ERROR OCCURS IN FIRST 20 RECORDS.
   20 WRITE(11,1200)
 1200 FORMAT(' '/' ','LOSS OF SYNCHRONIZATION WITHIN FIRST 20 RECORDS. A
     *UTOMATIC PROGRAM TERMINATION')
      STOP
C  COME HERE IF FIRST CALL TO SYNCRO.
   30 MEAN=(NSMREC-8)/SAMRAT
      IIFLAG=0
      GOTO 50
C  COMPUTE THE AVERAGE PERIOD FOR RECORDS CURRENTLY BEING PROCESSED.
   40 MEAN= (3*MEAN + DELTA)/4
C  STORE PRESENT RECTIM FOR USE DURING NEXT CALL TO SYNCRO.
   50 OLDTIM=RECTIM
      RETURN
   60 RATIO=DELTA/MEAN
      WRITE(11,1300) NBLOCK,DELTA,RATIO
 1300 FORMAT(' '/' ','NON-STANDARD TIME INTERVAL DETECTED IN RECORD NUMB
     *ER ',I5,'.'/' ','INTERVAL BETWEEN ADJACENT RECORDS =',F7.4,
     *' SECONDS, WHICH IS ABOUT ',F3.1,' STANDARD PERIODS.')
      GOTO 10
      END
      SUBROUTINE GETIME (KHRS,KMINS,SECS)
C
C  DECODES THE THREE TIME WORDS INTO HOURS, MINUTES AND SECONDS.
C
      COMMON/FETCH/TIMST,NCALL,NBLOCK,LHRS,LMINS,LSECS,LPTIME,
     *ARRAY
      COMMON/TAPE/SAMRAT,IWDNUM,NCHANL,NSMREC,NHDCAL,NID,CLRECL
      REAL MULT,MSECS,LENGTH
      INTEGER*2 ARRAY(1),K(16)
      LENGTH = (NSMREC - 8)/SAMRAT
```

```
      TIME = 1
      KWDNUM=2006
C  BREAK DOWN 16 BIT TIME WORD INTO SEPARATE BITS AND STORE
C  IN LENGTH 16 ARRAY "K".
   10 ITIME=ARRAY(KWDNUM)
      IST=1
      IF(ITIME.GE.0) GOTO 20
      K(16)=1
      ITIME=ITIME+32768
      IST=2
   20 DO 40 M=IST,16
         N=17-M
         ITEST=2**(N-1)
         IF(ITIME.GE.ITEST) GOTO 30
         K(N)=0
         GOTO 40
   30    K(N)=1
         ITIME=ITIME-ITEST
   40 CONTINUE
      IF(ITIME.NE.0) WRITE(6,1000)
 1000 FORMAT (1X,'ERROR IN READING TIME WORD. RESULT NOT ZERO')
      IF(KWDNUM-2007) 50,70,80
C  COMPUTE FRACTIONAL SECONDS.
   50 MULT=0.0001
      MSECS=0.0
      DO 60 IDX=1,13,4
         ITMSEC=8*K(IDX+3)+4*K(IDX+2)+2*K(IDX+1)+K(IDX)
         MSECS=MSECS+(ITMSEC*MULT)
         MULT=MULT*10.
   60 CONTINUE
      KWDNUM=2007
      GOTO 10
C  COMPUTE SECONDS AND MINUTES.
   70 SECS=40*K(7)+20*K(6)+10*K(5)+8*K(4)+4*K(3)+2*K(2)+K(1)+MSECS
      KMINS=40*K(15)+20*K(14)+10*K(13)+8*K(12)+4*K(11)+2*K(10)+K(9)
      KWDNUM=2008
      GOTO 10
C  COMPUTE HOURS.
   80 KHRS=20*K(6)+10*K(5)+8*K(4)+4*K(3)+2*K(2)+K(1)
C  SINCE TIME STORED IN LAST THREE WORDS OF RECORD IS THE STARTING
C  TIME OF THE NEXT RECORD, SUBTRACT DATA RECORD LENGTH TO CONVERT
C  TO RECORD START TIME.
      SECS=SECS-LENGTH
      RETURN
      END
      SUBROUTINE UTIME(TIM,UTIME1,UTIME2,UTIME3)
C
C  CONVERTS TIME IN SECONDS TO HOURS, MINUTES, AND SECONDS.
C  INPUT ARGUMENT:   TIM = TIME VALUE IN SECONDS.
C  OUTPUT ARGUMENTS: UTIME1 = HOURS PORTION OF TIME VALUE.
C                    UTIME2 = MINUTES PORTION OF TIME VALUE.
C                    UTIME3 = SECONDS PORTION OF TIME VALUE.
C
      TI1=TIM/3600.
```

```
      UTIME1=AINT(TI1)
      REM1=AMOD(TIM,3600.)
      ATI2=REM1/60.
      UTIME2=AINT(ATI2)
      UTIME3=AMOD(REM1,60.)
      RETURN
      END
      SUBROUTINE CONVER(TIMESH,TIMESM,TIMESS,TIMESA)
C
C  CONVERTS INPUT TIME VALUES (IN HOURS, MINS, & SECS) TO SECONDS.
C  INPUT ARGUMENTS: TIMESH = HOURS PORTION OF TIME VALUE.
C                   TIMESM = MINUTES PORTION OF TIME VALUE.
C                   TIMESS = SECONDS PORTION OF TIME VALUE.
C  OUTPUT ARGUMENT: TIMESA = TIME VALUE CONVERTED TO SECONDS.
C
      TIMESA=(TIMESH*3600.)+TIMESM*60.+TIMESS
      RETURN
      END
//GO.FT11F001 DD SYSOUT=A,DCB=(RECFM=FA,LRECL=133)
//GO.FT12F001 DD UNIT=TAPE,VOL=SER=PFT203,DISP=OLD,
// DCB=(RECFM=U,BLKSIZE=4016),LABEL=(1,BLP,,IN)
//GO.SYSIN DD *
 &FORM SAMRAT=5000.,IWDNUM=1,NCHANL=5,NSMREC=2008,NHDCAL=140,
 NID=4,CLRECL=1005,&END
1
1
00216
010
1652:00
     ## 18.1020 FINAL PROBE RUN ##
4.360
4.805     1652:26
4.722     1657:45
09
0.740     4.05E-11
1.440     4.05E-10
2.130     4.05E-09
2.860     4.05E-08
3.600     4.05E-07
4.340     4.05E-06
4.760     1.50E-05
5.120     4.05E-05
5.710     1.50E-04
 10.0      5.126
 20.0      9.411
 30.0     21.024
 40.0     35.095
 50.0     48.091
 60.0     60.089
 70.0     71.158
 80.0     81.264
 90.0     90.390
100.0     98.618
110.0    105.810
```

120.0	112.102
130.0	117.418
140.0	121.855
150.0	125.338
160.0	127.870
170.0	129.581
180.0	130.069
190.0	129.768
200.0	128.457
210.0	126.344
220.0	123.240
230.0	119.174
240.0	114.162
250.0	108.203
260.0	101.282
270.0	93.409
280.0	84.588
290.0	74.898
300.0	64.183
310.0	52.643
320.0	40.375
330.0	28.983
340.0	21.420
350.0	17.545
360.0	15.282
370.0	0.0
0.0	0.0

1652:51
1657:10
/*

APPENDIX II. Listing of CDC Fortran IV program SWEEP.

The first ten lines of this appendix are Cyber Operating System Commands used to fetch the probe current versus time output of program WPROBE, save it, compile and execute program SWEEP. The trajectory file, TRJ20, consists of the trajectory cards used as input to program WPROBE. GCSTEKT contains the graphics subroutines to plot on a tektronix 4006, 4010 or 4112 terminal. File GCSFNT (GCSfont) is returned because it is not needed to execute this program. Memory space is saved and loading time is reduced if GCSFNT is returned.

```
FETCH,PC20X <CR>
RENAME,PC20=PC20X <CR>
SAVE,PC20 <CR>
USE,SWEEP,TRJ20 <CR>
GET,SWEEP,TRJ20 <CR>
GRAB,GCSTEKT/F <CR>
RETURN,GCSFNT <CR>
R.FTN,I=SWEEP,L=0 <CR>
LGO,,,PC20,PC20S,DIG20,TRJ20 <CR>
SAVE,PC20S,DIG20 <CR>
```

```
      PROGRAM SWEEP(INPUT,OUTPUT,PCIN,PCOUT,DIGNOS,TRJTRY,
     *TAPE1=PCIN,TAPE2=PCOUT,TAPE3=DIGNOS,TAPE4=TRJTRY,
     *TAPE5=INPUT)
C
C THIS PROGRAM IS DESIGNED FOR USE ON THE OUTPUT FROM PROGRAM
C WPROBE. THE OUTPUT FILE FROM SWEEP WILL BE IDENTICAL WITH THE INPUT
C FILE FOR ALL PERIODS WHEN THE LANGMUIR PROBE WAS HELD AT A CONSTANT
C POTENTIAL. WHEN THE LANGMUIR PROBE IS SWEPT,THE PROBE CURRENT VALUE
C ON THE INPUT FILE IS REPLACED BY THE VALUE 2.0 ON THE
C OUTPUT FILE. THE OUTPUT FILE MAY BE PLOTTED WITH PROGRAM
C EDPLOT.  EDPLOT IS DESIGNED NOT TO PLOT THE FLAG
C VALUE OF 2.0. THE PLOT PRODUCED WILL HAVE BLANK SPACES IN
C THE PROBE CURRENT CURVE WHEREVER A SWEEP OCCURED, OR THE OUTPUT FILE
C MAY BE USED WITH PROGRAM PCTOED.
C
C ALSO INCLUDED IS A ROUTINE FOR PLOTTING UP TO 2000 POINTS
C ON THE TERMINAL FOR INSPECTION.
C DATA CAN BE EITHER FROM THE INPUT TO SWEEP (PCIN) OR THE OUTPUT
C FROM SWEEP (PCOUT).
C
C AUTHORS: R.K.ZIMMERMAN, UNIVERSITY OF ILLINOIS - AERONOMY LABORATORY
C          M.K.MCINERNEY, UNIVERSITY OF ILLINOIS - AERONOMY LABORATORY
C
      COMMON/TRJ/T,ALTI,TRJCAL
      REAL AINCR,ALT(6000),ALTI(60),ALTMAX,BTIME,MIN,PC(6000),PCMAX,
     *PCMAX1,PCMAX2,PCMIN,PCMIN1,PCMIN2,PMAX,PMIN,RATIO,RLCHNG,SLOPE,
     *T(60),TIM(1200),XMIN,XMAX,XSIZE,YMIN,YMAX,YSIZE,Z
      INTEGER ARRAY(200),INDEX,K,KK,LL,MASK(200,2),MAXI,MAXJ,MM,NN,
     *OVRPLT,PCFILE,PLST,PRENDX,RR,TMFILE
```

```
C
C  INITIALIZE VARIABLES.
C
      DATA FF/0007403340014B/
      DATA PC/6000*1.0/
      PLST=0
      RLCHNG=-1.0
      T(1)=0.0
      ALTI(1)=0.0
      PCFILE=0
      TMFILE=0
      REWIND 3
      REWIND 4
      CALL USTART
      CALL DEVICE(XSIZE,YSIZE)
      CALL UMOVE(0.0,100.0)
      CALL UFLUSH
      CALL UALPHA
      WRITE 1000
 1000 FORMAT(" "/" "/" ","* YOU HAVE TWO CHOICES."/" ",
     *" 1 - TO GO DIRECTLY TO THE PLOT ROUTINE AND PLOT"/" ",
     *"     DATA FROM EITHER PCIN OR PCOUT."/" ",
     *" 2 - TO GO DIRECTLY TO PROGRAM SWEEP AND SET THE"/" ",
     *"     AFOREMENTIONED PROGRAM LOOSE ON FILE PCIN."/" ",
     *" TYPE 1 OR 2.")
   10 READ *, NN
      IF(EOF(5).NE.0) GOTO 10
      IF(NN.EQ.1) GOTO 270
      IF(NN.EQ.2) GOTO 20
      GOTO 10
   20 RLCHNG=0.0
      WRITE 1100, FF
 1100 FORMAT(R6)
      CALL UWAIT(2.0)
      WRITE 2700
   30 READ 3400,A,B,C
      IF(EOF(5).NE.0) GOTO 30
   40 IF(RLCHNG.EQ.-1.0) GOTO 20
      WRITE 2900
C
C RLCHNG STANDS FOR RELATIVE CHANGE.
C THE RATIO OF TWO PROBE CURRENT VALUES IS COMPUTED.
C THE DEVIATION OF THIS RATIO FROM 1 IS THE RELATIVE CHANGE.
C (I.E. RLCHNG=ABS((PC(N-1)/PC(N))-1).)
C FOR DATA WITHOUT LARGE GRADIENTS, RLCHNG CAN BE 0.02.
C FOR DATA WITH LARGE GRADIENTS, RLCHNG COULD BE 0.40.
C
   50 READ 3000, RLCHNG
      IF(EOF(5).NE.0) GOTO 50
      WRITE 1100,FF
      CALL UWAIT(2.0)
      WRITE 3100,RLCHNG,A,B,C
      REWIND 2
      WRITE(3,2800) A,B,C
```

```
      WRITE(3,3200) RLCHNG
C
C INPUT OF TIME/CURRENT CARDS.
C
      NN=1
      IF(PCFILE.EQ.NN) GOTO 60
      CALL PREAD(TIM,PC,NN,MAXI,MAXJ,PCFILE)
      IF(MAXI.NE.0) GOTO 60
      PRINT*, "* THERE IS NO DATA IN FILE PCIN,"
      PRINT*, "  EXECUTION IS TERMINATED."
      STOP 1
C
C PRINT NUMBER OF DATA LINES READ IN.
C
   60 WRITE(3,2000) MAXI, MAXJ
      PRINT*, "* AT WHAT TIME DO YOU WANT TO"
      PRINT*, "  BEGIN REMOVING SWEEPS?"
   70 READ*, BTIME
      IF(EOF(5).NE.0) GOTO 70
      IF(BTIME.LE.TIM(MAXI)) GOTO 80
      PRINT*, "* THE TIME THAT YOU ENTERED IS GREATER THAN THE"
      PRINT*, "  LAST DATA TIME.", TIM(MAXI)
      PRINT*, "  TRY AGAIN."
      GOTO 70
   80 PRINT(3,*) "PROGRAM SWEEP IS BEGINNING AT TIME =",BTIME
C
C A SWEEP HAS BEEN LOCATED IF SLOPE IS GREATER THAN 3.0.
C
      J=5
      K=0
   90 J=J+1
      IF(TIM(INT(J/5.0)+1).LT.BTIME) GOTO 90
      IF(J.GT.MAXJ-5) GOTO 130
      IF(PC(J).EQ.1.0) GOTO 90
      SLOPE=PC(J)/PC(J+1)
      IF(SLOPE.GT.3.0) GOTO 100
      GOTO 90
C
C FIND THE SWEEP-CENTERED PROBE CURRENT BY LOOKING FOR THE SMALLEST PC.
C
  100 J=J+1
      IF(J.GT.MAXJ-5) GOTO 120
      MIN=PC(J)
      IF(MIN.GT.PC(J+1)) GOTO 100
      IF(MIN.LT.((PC(J-5)+PC(J+5))/40.0)) GOTO 110
      WRITE(3,2100) J
C
C THE PRESENT VALUE OF PROBE CURRENT IS EVIDENTLY NOT A SWEEP MINIMUM.
C PASS IT BY AND CONTINUE THE SEARCH FOR LANGMUIR PROBE SWEEPS.
C
      GOTO 90
C
C COME HERE IF THE CENTER OF A SWEEP HAS BEEN LOCATED.
C
```

```
  110 K=K+1
      ARRAY(K)=J
C
C RETURN TO FIND OTHER SWEEPS.
C
      GOTO 90
C
C ARRAY NOW CONTAINS THE INDICES OF THE SWEEP-CENTERED PROBE CURRENT.
C
C THE POINTS PRECEEDING THE SWEEP CENTER MUST BE
C EXAMINED TO SEE IF THEY ARE CONTAINED IN THE
C SWEEP PERIOD.
C
  120 K=K+1
      ARRAY(K)=J
  130 DO 210 M=1,K
          II=ARRAY(M)
          MASK(M,1)=II
          DO 150 MI=1,10
             RATIO=PC(II-MI-1)/PC(II-MI)
             IF(ABS(RATIO-1.0).GT.RLCHNG) GOTO 140
             RATIO=PC(II-MI)/PC(II)
             IF(RATIO.GT.20.0) GOTO 170
  140        MASK(M,1)=II-MI
  150     CONTINUE
C
C THE ABOVE DO-LOOP SHOULD NEVER RUN TO COMPLETION. IF IT DOES,SOMETHING
C IS FOUL. THE DATA IS DUMPED.
C
          MASK(M,1)=II
          WRITE(3,2200)
          DO 160 ID=1,11
             IIO=II-(12-ID)
             WRITE(3,2300)IIO, PC(IIO)
  160     CONTINUE
          WRITE(3,2400)II,PC(II)
C
C THE POINTS FOLLOWING THE SWEEP CENTER MUST BE
C EXAMINED TO SEE IF THEY ARE CONTAINED IN THE
C SWEEP PERIOD.
C
  170     MASK(M,2)=II
          DO 190 NI=1,10
             RATIO=PC(II+NI+1)/PC(II+NI)
             IF(ABS(RATIO-1.0).GT.RLCHNG) GOTO 180
             RATIO=PC(II+NI)/PC(II)
             IF(RATIO.GT.20.0) GOTO 210
  180        MASK(M,2)=II+NI
  190     CONTINUE
C
C THE ABOVE DO-LOOP SHOULD NEVER RUN TO COMPLETION. IF IT
C DOES, SOMETHING IS FOUL. THE DATA IS DUMPED FOR INSPECTION.
C
          MASK(M,2)=II
```

```
         WRITE(3,2200)
         WRITE(3,2400) II, PC(II)
         DO 200 ID=1,11
            IIO=II+ID
            WRITE(3,2300) IIO, PC(IIO)
  200    CONTINUE
  210 CONTINUE
C
C REPLACE EACH PC CONTAINED IN A SWEEP PERIOD WITH THE FLAG VALUE 2.0.
C
      WRITE(3,2500)
      DO 240 M=1,K
         ISTART=MASK(M,1)
         ISTOP=MASK(M,2)
         DO 220 KI=ISTART,ISTOP
            PC(KI)=2.0
  220    CONTINUE
         ICARD1=((ISTART-1)/5)+1
         ICARD2=((ISTOP-1)/5)+1
C
C PRINT CARD IMAGES.
C
         WRITE(3,2600)
         DO 230 KI=ICARD1,ICARD2
            J1=5*KI-4
            J2=5*KI
            WRITE(3,3500) TIM(KI),(PC(N),N=J1,J2)
  230    CONTINUE
  240 CONTINUE
C
C CREATE OUTPUT FILE.
C
      DO 250 KI=1,MAXI
         J1=5*KI-4
         J2=5*KI
         WRITE(2,3500) TIM(KI),(PC(N),N=J1,J2)
  250 CONTINUE
      PCFILE=2
C
C  HERE BEGINS THE PLOT SECTION OF THIS PROGRAM.
C
      PRINT*, "*  DO YOU WISH TO PLOT ANY DATA?"
  260 READ 1200,H
 1200 FORMAT(1A1)
      IF(EOF(5).NE.0) GOTO 260
      IF(H.EQ."N         ") GOTO 810
      IF(H.EQ."Y         ") GOTO 270
      GOTO 260
  270 PRINT*, "* DO YOU WANT TO PLOT DATA FROM PCIN"
      PRINT*, "  OR FROM PCOUT?"
      PRINT*, " TYPE 1 FOR PCIN."
      PRINT*, " TYPE 2 FOR PCOUT."
  280 READ*, NN
      IF(EOF(5).NE.0) GOTO 280
```

```
      IF(NN.NE.1.AND.NN.NE.2) GOTO 270
      RR=1
      IF(NN.EQ.1) RR=0
      IF(RLCHNG.EQ.-1.0) RR=0
C
C  READ VALUES OF EITHER PCIN OR PCOUT INTO ARRAY PC.
C
      IF(PCFILE.EQ.NN) GOTO 300
      CALL PREAD(TIM,PC,NN,MAXI,MAXJ,PCFILE)
      IF(MAXI.NE.0) GOTO 300
      IF(PCFILE.EQ.1) GOTO 290
      PRINT*, "* THERE IS NO DATA IN PCOUT,TRY A DIFFERENT FILE."
      GOTO 270
  290 PRINT*, "* THERE IS NO DATA IN FILE PCIN,"
      PRINT*, "  EXECUTION IS TERMINATED."
      STOP 2
C
C  DETERMINE THE TIME IN SECONDS BETWEEN EACH PROBE CURRENT POINT.
C
  300 IF(TMFILE.EQ.NN) GOTO 380
      AINCR=(TIM(2)-TIM(1))/5.0
C
C  ASSIGN A TIME TO EACH PROBE CURRENT.
C
      DO 310 I=1,MAXI
         ALT((I*5)-4)=TIM(I)
         DO 310 J=1,4
            ALT((I*5)-4+J)=TIM(I)+AINCR*J
  310 CONTINUE
C
C  INPUT TRAJECTORY DATA FROM TAPE 4=TRJTRY.
C  NOTE:  ARRAY LOCATION T(1) IS USED AS A FLAG TO INDICATE IF
C         THE TRAJECTORY DATA HAS BEEN READ IN.
C
      IF(T(1).EQ.1.0) GOTO 360
      I=1
  320 I=I+1
      READ(4,1300) T(I),ALTI(I)
      IF(EOF(4)) 330,320
 1300 FORMAT(F7.1,F10.3)
  330 IF(I.NE.2) GOTO 350
      PRINT*, "* THERE IS NO DATA IN THE TRAJECTORY FILE."
      PRINT*, "  DO YOU WISH TO CONTINUE WITH THE SWEEP"
      PRINT*, "  REMOVING ROUTINE AND NOT DO ANY PLOTTING?"
      PRINT*, "TYPE Y TO GO TO SWEEP"
      PRINT*, "TYPE N TO TERMINATE EXECUTION."
  340 READ 1200,H
      IF(EOF(5).NE.0) GOTO 340
      IF(H.EQ."Y         ") GOTO 40
      IF(H.EQ."N         ".AND.RLCHNG.EQ.-1.0) STOP 3
      IF(H.EQ."N         ") GOTO 810
      GOTO 340
  350 T(1)=1.0
C
```

```
C  ASSIGN AN ALTITUDE TO EACH PROBE CURRENT AND DETERMINE THE MINIMUM
C  AND MAXIMUM ALTITUDES OF THE DATA.
C
  360 TRJCAL=0
      ALTMAX=0.0
      DO 370 I=1,MAXJ
         CALL TRAJ(ALT(I),ALT(I))
         ALTMAX=AMAX1(ALTMAX,ALT(I))
         IF(ALTMAX.EQ.ALT(I)) MM=I
  370 CONTINUE
      TMFILE=NN
  380 IF(RLCHNG.EQ.-1.0) GOTO 420
      PRINT*, "* DO YOU WISH TO SEE THE ALTITUDES AT"
      PRINT*, "  WHICH SWEEPS OCCURRED?"
  390 READ 1200,H
      IF(EOF(5).NE.0) GOTO 390
      IF(H.EQ."N         ") GOTO 420
      IF(H.EQ."Y         ") GOTO 400
      GOTO 390
C
C  MM IS THE INDEX OF THE MAXIMUM ALTITUDE.
C
C  WRITE THE VALUES AT WHICH A SWEEP OCCURRED.
C
  400 WRITE 1100, FF
      CALL UWAIT(2.0)
      DO 410 I=1,K,7
         IF(ARRAY(I).GT.MM) GOTO 430
         WRITE 1400, (ALT(ARRAY(I+J-1)),J=1,7)
  410 CONTINUE
 1400 FORMAT(" ",7F10.3)
C
C  INPUT VALUES OF STARTING AND ENDING ALTITUDES FOR
C  PLOT AND CHECK THEIR VALIDITY.
C
  420 WRITE 1100,FF
      CALL UWAIT(2.0)
  430 PRINT*, "* TO PLOT ASCENDING DATA, HAVE STARTING ALTITUDE"
      PRINT*, "  LESS THAN ENDING ALTITUDE."
      PRINT*, "   "
      PRINT*, "* TO PLOT DESCENDING DATA, HAVE STARTING ALTITUDE"
      PRINT*, "  GREATER THAN ENDING ALTITUDE."
      PRINT*, "   "
      WRITE 1500, ALT(1),ALT(MAXJ)
 1500 FORMAT(" ","* THE MINIMUM DATA ALTITUDES ARE ",F7.3," AND"/F7.3,
     *" "," FOR ASCENT AND DESCENT, RESPECTIVELY.")
      PRINT*, "   "
      WRITE 1600,ALTMAX
 1600 FORMAT(" ","* THE MAXIMUM DATA ALTITUDE IS ",F7.3,".")
      PRINT*, "   "
      PRINT*, "* WHAT IS THE STARTING PLOT ALTITUDE? (XXX.XXX)"
  440 READ*, PMIN
      IF(EOF(5).NE.0) GOTO 440
      IF(PMIN.LE.0.0) GOTO 440
```

```
      PRINT*, "   "
      PRINT*, "* WHAT IS THE ENDING PLOT ALTITUDE? (XXX.XXX)"
  450 READ*, PMAX
      IF(EOF(5).NE.0) GOTO 450
      IF(PMAX.LE.0.0) GOTO 450
C
C   FIND OUT IF THE PREVIOUS PLOT AND THE PRESENT PLOT ARE
C   TO BE SUPERIMPOSED (PLOTTED ON TOP OF EACH OTHER).
C
      IF(PLST.EQ.0) GOTO 470
      IF(XSIZE.LT.7.75.OR.8.5.LT.XSIZE) GOTO 470
      WRITE 1100, FF
      CALL UWAIT(2.0)
      PRINT*, "* DO YOU WANT TO HAVE THE PRESENT PLOT AND"
      PRINT*, "  THE PREVIOUS PLOT PLOTTED ON THE SAME"
      PRINT*, "  AXIS ?"
  460 READ 1200, H
      IF(EOF(5).NE.0) GOTO 460
      IF(H.NE.1HN.AND.H.NE.1HY) GOTO 460
      IF(H.EQ.1HN) OVRPLT=0
      IF(H.EQ.1HY) OVRPLT=1
  470 IF(PMAX.GT.PMIN) GOTO 520
C
C  FOR DESCENT   DETERMINE THE INDEX OF THE MAXIMUM PLOT ALTITUDE
C                IN THE ARRAY ALT
C
      DO 480 I=MM,MAXJ
         LL=I
         Z=AMAX1(ALT(I),PMIN)
         IF(Z.EQ.PMIN) GOTO 490
  480 CONTINUE
C
C  DEFINE THE MAXIMUM ALTITUDE ON VERTICAL AXIS
C
  490 YMAX=AINT(ALT(LL))+1.0
      YMAX=AMAX1(YMAX,PMIN)
C
C  FOR DESCENT  DETERMINE THE INDEX OF THE MINIMUM PLOT ALTITUDE
C               IN THE ARRAY ALT
C
      DO 500 I=LL,MAXJ
         KK=I-1
         Z=AMAX1(ALT(I),PMAX)
         IF(Z.EQ.PMAX) GOTO 510
         IF((KK-LL+1).EQ.2000) GOTO 560
  500 CONTINUE
C
C  DEFINE THE MINIMUM ALTITUDE ON VERTICAL AXIS
C
  510 YMIN=AINT(ALT(KK))
      YMIN=AMIN1(YMIN,PMAX)
      GOTO 580
C
C FOR ASCENT    DETERMINE THE INDEX OF THE MINIMUM PLOT ALTITUDE
```

```
C                IN THE ARRAY ALT
C
  520 DO 530 I=1,MM
         LL=I
         Z=AMAX1(ALT(I),PMIN)
         IF(Z.EQ.ALT(I)) GOTO 540
  530 CONTINUE
C
C  DEFINE MINIMUM VALUE ON VERTICAL AXIS
C
  540 YMIN=AINT(ALT(LL))
      YMIN=AMIN1(YMIN,PMIN)
C
C  FOR ASCENT   DETERMINE THE INDEX OF THE MAXIMUM PLOT ALTITUDE
C                IN THE ARRAY ALT
C
      DO 550 I=LL,MM
         KK=I-1
         Z=AMAX1(ALT(I),PMAX)
         IF(Z.EQ.ALT(I)) GOTO 570
         IF((KK-LL+1).EQ.2000) GOTO 560
  550 CONTINUE
      GOTO 570
  560 WRITE 1100,FF
      CALL UWAIT(2.0)
      WRITE 1700, ALT(KK)
 1700 FORMAT(" ","* YOU HAVE ATTEMPTED TO PLOT MORE THAN THE MAXIMUM",/,
     *"  OF 2000 POINTS.  THE ENDING PLOT ALTITUDE HAS BEEN CHANGED",/,
     *"  TO ",F7.3," SO THAT ONLY 2000 POINTS WILL BE PLOTTED.",/,
     *"  PRESS RETURN TO CONTINUE.")
      PAUSE
      IF(PMAX.LE.PMIN) GOTO 510
C
C  DEFINE THE MAXIMUM PLOT VALUE ON VERTICAL AXIS
C
  570 YMAX=AINT(ALT(KK))+1.0
      YMAX=AMAX1(YMAX,PMAX)
C
C  LL IS THE INDEX OF THE STARTING ALTITUDE
C  KK IS THE INDEX OF THE ENDING ALTITUDE
C
  580 IF((KK-LL).GE.4) GOTO 590
      WRITE 1100,FF
      CALL UWAIT(2.0)
      PRINT*, "* YOU MUST PLOT AT LEAST 5 DATA POINTS.  TRY AGAIN."
      GOTO 420
C
C  FIND THE MINIMUM AND MAXIMUM VALUES OF PROBE CURRENT OVER THE
C  INTERVAL TO BE PLOTTED.
C
  590 IF(OVRPLT.EQ.1) GOTO 720
      PCMIN=1.0E10
      PCMAX=0.0
      DO 600 I=LL,KK
```

```
      IF(PC(I).EQ.1.0) GOTO 600
      IF(PC(I).EQ.2.0) GOTO 600
      PCMIN=AMIN1(PCMIN,PC(I))
      PCMAX=AMAX1(PCMAX,PC(I))
      PRENDX=I
  600 CONTINUE
C
C  REQUEST MIN AND MAX PROBE CURRENT VALUES AS POWERS OF 10.
C
      PCMIN2=ALOG10(PCMIN)
      PCMAX2=ALOG10(PCMAX)
      PCMIN2=AINT(PCMIN2)-1.0
      PCMAX2=AINT(PCMAX2)
      PRINT*, "* DO YOU WANT TO SCALE THE CURRENT AXIS"
      PRINT*, "  YOURSELF OR DO YOU WISH TO HAVE THE"
      PRINT*, "  PLOT ROUTINE SCALE IT?"
      PRINT*, " TYPE YES TO SCALE YOURSELF."
      PRINT*, " TYPE NO TO HAVE PLOT ROUTINE SCALE."
  610 READ 1200, H
      IF(EOF(5).NE.0) GOTO 610
      IF(H.EQ."N         ") GOTO 700
      IF(H.NE."Y         ") GOTO 610
  620 WRITE 1100,FF
      CALL UWAIT(2.0)
      PRINT*, "* WHAT IS THE MINIMUM CURRENT AXIS VALUE"
      PRINT*, "  AS A POWER OF 10,I.E. 10**X."
      PRINT*, "  ENTER X."
  630 READ*, PCMIN1
      IF(EOF(5).NE.0) GOTO 630
      IF(PCMIN1.LE.PCMIN2) GOTO 650
      PRINT*, "* WARNING *  THE MINIMUM CURRENT VALUE IS ",PCMIN
      PRINT*, "  DO YOU WISH TO CONTINUE ANYWAY?"
  640 READ 1200, H
      IF(EOF(5).NE.0) GOTO 640
      IF(H.EQ."N         ") GOTO 620
      IF(H.EQ."Y         ") GOTO 650
      GOTO 640
  650 PCMIN2=PCMIN1
  660 PRINT*, "* ENTER THE MAXIMUM CURRENT AXIS VALUE"
      PRINT*, "  AS A POWER OF 10, I.E. 10**X."
  670 READ*, PCMAX1
      IF(EOF(5).NE.0) GOTO 670
      IF(PCMAX1.GE.PCMAX2) GOTO 690
      PRINT*, "* WARNING *  THE MAXIMUM CURRENT VALUE IS",PCMAX
      PRINT*, "  DO YOU WISH TO CONTINUE ANYWAY?"
  680 READ 1200, H
      IF(EOF(5).NE.0) GOTO 680
      IF(H.EQ."Y         ") GOTO 690
      IF(H.EQ."N         ") GOTO 660
      GOTO 680
  690 PCMAX2=PCMAX1
  700 XMIN=10**PCMIN2
      XMAX=10**PCMAX2
C
```

```
C  PLOT.
C
      CALL URESET
      CALL UERASE
      PLST=1
      CALL UDAREA(0.0,XSIZE,0.0,YSIZE)
      IF(RR.EQ.0) GOTO 710
      CALL USET("TEXT")
      CALL UPRINT(0.0,0.0,"  RLCHNG=;")
      CALL UPRNT1(RLCHNG,"REAL")
      CALL UDAREA(0.0,XSIZE,0.125,YSIZE)
  710 CALL USET("LOGXAXIS")
      CALL USET("XLOGARITHMIC")
      CALL USET("XBOTHLABELS")
      CALL USET("YBOTHLABELS")
      CALL USET("OWNSCALE")
      CALL UPSET("XLABEL","PROBE CURRENT;")
      CALL UPSET("YLABEL","ALTITUDE;")
      CALL UPSET("TICY",5.0)
      CALL UWINDO(XMIN,XMAX,YMIN,YMAX)
      CALL UAXIS(XMIN,XMAX,YMIN,YMAX)
  720 INDEX=LL
      PRENDX=INDEX
      IF(PC(INDEX).EQ.1.0.OR.PC(INDEX).EQ.2.0) GOTO 740
      CALL UMOVE(PC(INDEX),ALT(INDEX))
      PRENDX=INDEX
      INDEX=INDEX+1
      IF(INDEX.GT.KK) GOTO 770
  730 IF(PC(INDEX).EQ.1.0.OR.PC(INDEX).EQ.2.0) GOTO 740
      CALL UPEN(PC(INDEX),ALT(INDEX))
      PRENDX=INDEX
      INDEX=INDEX+1
      IF(INDEX.GT.KK) GOTO 770
      GOTO 730
  740 CALL UMOVE(PC(PRENDX),ALT(INDEX))
      IF(PC(INDEX+1).NE.1.0.AND.PC(INDEX+1).NE.2.0) GOTO 750
      INDEX=INDEX+1
      IF(INDEX.GT.KK) GOTO 770
      GOTO 740
  750 INDEX=INDEX+1
      IF(INDEX.GT.KK) GOTO 770
      IF(PC(INDEX+1).EQ.1.0.OR.PC(INDEX+1).EQ.2.0) GOTO 760
      CALL UMOVE(PC(INDEX),ALT(INDEX))
      PRENDX=INDEX
      INDEX=INDEX+1
      IF(INDEX.GT.KK) GOTO 770
      GOTO 730
  760 CALL USET("NSYMBOL")
      CALL UPSET("SZMARKER",.08)
      CALL UPSET("SYMBOL",(2.0*FLOAT(PCFILE)))
      CALL UPEN(PC(INDEX),ALT(INDEX))
      CALL USET("LINE")
      PRENDX=INDEX.
      INDEX=INDEX+1
```

```
      IF(INDEX.GT.KK) GOTO 770
      GOTO 740
  770 CALL UBELL
      CALL UHOME
      CALL UFLUSH
      CALL UPAUSE
C
C REQUEST ANOTHER PLOT.
C
      CALL UALPHA
      WRITE 1100, FF
      CALL UWAIT(2.0)
      PRINT*, "* DO YOU WISH TO PLOT AGAIN WITH DIFFERENT PARAMETERS?"
  780 READ 1200,H
      IF(EOF(5).NE.0) GOTO 780
      IF(H.EQ."Y          ") GOTO 270
      IF(H.EQ."N          ") GOTO 790
      GOTO 780
  790 PRINT*, "* DO YOU WISH TO GO BACK AND RERUN SWEEP"
      PRINT*, "  WITH A DIFFERENT VALUE FOR RLCHNG?"
  800 READ 1200,H
      IF(EOF(5).NE.0) GOTO 800
      IF(H.EQ."Y          ") GOTO 40
      IF(H.EQ."N          ") GOTO 810
      GOTO 800
  810 WRITE 3300
      CALL UEND
C
 2000 FORMAT(" NUMBER OF LINES READ = ",I4,/" NUMBER OF PROBE CURRENT VA
     *LUES READ = ",I4)
 2100 FORMAT(" DATA IS PARTICULARLY NOISY ABOUT PC(",I4,").")
 2200 FORMAT(" "/" THIS DATA IS FOUL (NOISY).")
 2300 FORMAT(" PC(",I4,")=",E13.4)
 2400 FORMAT(" PC(",I4,")=",E13.4," IS THE SWEEP CENTER.")
 2500 FORMAT(" "/" "/" THE FOLLOWING LINES ARE CHANGED:")
 2600 FORMAT(" ")
 2700 FORMAT(" "/" "/10X,
     *            " SSSSS   W     W   EEEEEE   EEEEEE   PPPPPP",11X,"AERO
     *NOMY"/10X,"S",8X,"W     W   E",8X,"E",8X,"P     P"/10X,"S",8X,"W
     *   W   E",8X,"E",8X,"P     P",8X,"LABORATORY"/10X," SSSS   W  W
     *W   EEEEEE   EEEEEE   PPPPPP"/15X,"S   W W W W   E",8X,"E",8X,"P",
     *18X,"ROCKET"/15X,"S   WW   WW   E",8X,"E",8X,"P"/10X,"SSSSS    W
     *   W   EEEEEE   EEEEEE   P",17X,"PROGRAM"/" "/" "/"* ENTER ROCKET
     *TYPE AND NUMBER BELOW.")
 2800 FORMAT(1X,3A10)
 2900 FORMAT(" "/"* THE PURPOSE OF PROGRAM SWEEP IS TO REMOVE THE PORTIO
     *NS OF THE PROBE"/"CURRENT PROFILE THAT OCCURRED WHEN THE LANGMUIR
     *PROBE WAS BEING VOLTAGE"/"SWEPT.  THE OCCURANCE OF A SWEEP IS DETE
     *CTED BY EXAMINING THE RELATIVE"/"CHANGE BETWEEN CONSECUTIVE DIGITI
     *ZED VALUES OF PROBE CURRENT."/" "/"(I.E. THE RATIO OF TWO PROBE CU
     *RRENT VALUES IS COMPUTED.  THE DEVIATION"/" OF THIS RATIO FROM 1 I
     *S THE RELATIVE CHANGE:"/41X,"RLCHNG=ABS((PC(N-1)/PC(N))-1).)"/" "/
     *"* FOR DATA TAKEN UNDER QUIET CONDITIONS THE RELATIVE CHANGE PARAM
     *ETER"/"CAN BE 0.02.  A GOOD FIRST-SHOT VALUE IS RELATIVE CHANGE = 0
```

```
     *.05."/"FOR DATA TAKEN UNDER NOISY (DISTURBED) CONDITIONS RELATIVE
     *CHANGE CAN"/"BE AS HIGH AS 0.40 FOR PROPER DISCRIMINATION."/"SPECI
     *FY THE RELATIVE CHANGE PARAMETER BELOW (X.XX).")
 3000 FORMAT(F4.2)
 3100 FORMAT(" "/"O.K., RELATIVE CHANGE =" ,F4.2," FOR ",3A10)
 3200 FORMAT(" RELATIVE CHANGE PARAMETER =",F4.2)
 3300 FORMAT(" "/  "* PROGRAM SWEEP IS NOW COMPLETE. PLEASE CHECK FILE D
     *IGNOS FOR DIAGNOSTIC"/  "REMARKS BEFORE USING THE OUTPUT DATA IN F
     *ILE PCOUT. (EACH LINE OF DATA"/"THAT HAS BEEN ALTERED BY SWEEP IS
     *REPRODUCED IN DIGNOS.)."  /  " "/"*** IMPORTANT NOTICE ***   FILES
     * DIGNOS AND PCOUT ARE LOCAL FILES AND"  /"WILL VANISH IF YOU DO NO
     *T SAVE (OR REPLACE) THEM BEFORE SIGNING OFF.")
 3400 FORMAT(3A10)
 3500 FORMAT(F7.3,5E13.4)
      STOP 4
      END
      SUBROUTINE TRAJ(TIME,HEIGHT)
C
C COMPUTES INTERPOLATED ALTITUDE VALUE FOR ANY TIME AFTER LAUNCH.
C MODIFIED FOR USE WITH PROGRAM SWEEP. ALTITUDE AND TIME VALUES
C AT 10 SECOND INTERVALS MUST BE PROVIDED VIA ARRAYS T AND ALTI
C FOR ENTIRE PERIOD OF FLIGHT. SWEEP READS THESE ARRAYS FROM FILE
C TRJTRY.
C INPUT  ARGUMENT:   TIME = TIME IN SECONDS AFTER LAUNCH.
C OUTPUT ARGUMENT: HEIGHT = ALTITUDE IN KILOMETERS.
C
      COMMON/TRJ/T,ALTI,TRJCAL
      REAL ALTI(60),T(60)
      INCR=0
C
C IF FIRST CALL TO TRAJ, INITIALIZE VARIABLES. IF NOT SKIP TO 20.
C
      IF(TRJCAL.GT.0) GOTO 20
      I=4
      TRJCAL=1
   10 INCR=1
      IM1=I-1
      IM2=I-2
   20 IF(TIME.LT.T(IM2))WRITE 1000,TIME
 1000 FORMAT(" "/"TIME LESS THAN LOWEST TRAJECTORY POINT,  TIME=",F7.3)
C
C LOCATE TIME VALUES WHICH BRACKET PRESENT INPUT TIME VALUE.
C
      IF(TIME.LE.T(I)) GOTO 30
      I=I+1
      GOTO 10
C
C IF PRESENT TIME VALUE IS IN SAME INTERVAL AS PREVIOUS ONE,
C COMPUTE ALTITUDE WITH OLD COEFFICIENTS. IF NOT, COMPUTE NEW
C COEFFICIENTS FIRST.
C
   30 IF(INCR.EQ.1) GOTO 50
   40 HEIGHT=A*TIME*TIME+B*TIME+C
      RETURN
```

```
   50 BRAC1=(T(I)-T(IM1))*(ALTI(IM1)-ALTI(IM2))
      BRAC2=(T(IM1)-T(IM2))*(ALTI(I)-ALTI(IM1))
      TOP=BRAC1-BRAC2
      BRAC1=(T(IM1)-T(IM2))*(T(I)*T(I)-T(IM1)*T(IM1))
      BRAC2=(T(I)-T(IM1))*(T(IM1)*T(IM1)-T(IM2)*T(IM2))
      BOTTOM=BRAC2-BRAC1
      A=TOP/BOTTOM
      B=(ALTI(IM1)-ALTI(IM2))-A*(T(IM1)*T(IM1)-T(IM2)*T(IM2))
      B=B/(T(IM1)-T(IM2))
      C=ALTI(IM2)-A*T(IM2)*T(IM2)-B*T(IM2)
      GOTO 40
      END
      SUBROUTINE PREAD(TIM,PC,NN,MAXI,MAXJ,PCFILE)
C
C  SUBROUTINE PREAD READS DATA FROM EITHER PCIN OR PCOUT
C  INTO TIM AND PC.
C
      REAL PC(6000),TIM(1200)
      INTEGER MAXI,MAXJ,NN,PCFILE
      REWIND NN
      PCFILE=NN
      I=0
      J=0
   10 I=I+1
      J=J+5
      JL=J-4
      READ(NN,1000) TIM(I),(PC(K),K=JL,J)
      IF(EOF(NN)) 20,10
 1000 FORMAT(F7.3,5E13.4)
   20 MAXI=I-1
      MAXJ=J-5
      RETURN
      END
      SUBROUTINE DEVICE(MAXXDIM,MAXYDIM)
C
C THIS SUBROUTINE DETERMINES WHICH DEVICE IS BEING USED FOR THE
C PLOTTING AND SETS THE HORIZONTAL AND VERTICAL SIZES ACCORDINGLY.
C
      REAL LIMIT(8),MAXXDIM,MAXYDIM
      CALL USTUD(LIMIT)
      IF(LIMIT(6).LT.6.0) GOTO 10
      IF(LIMIT(6).LT.8.0) GOTO 20
      IF(LIMIT(6).LT.11.0) GOTO 30
      IF(LIMIT(6).LT.14.0) GOTO 40
      IF(LIMIT(6).LT.15.0) GOTO 50
C
C DEVICE IS ALPH OR ADDR.
C
   10 MAXXDIM=7.09
      MAXYDIM=5.74
      RETURN
C
C DEVICE IS TEKT.
C
```

```
   20 MAXXDIM=7.49
      MAXYDIM=5.71
      RETURN
C
C DEVICE IS ZETA.
C
   30 MAXXDIM=8.0
      MAXYDIM=9.99
      RETURN
C
C DEVICE IS PRNT.
C
   40 MAXXDIM=12.99
      MAXYDIM=7.37
      RETURN
C
C DEVICE IS TK14.
C
   50 MAXXDIM=14.33
      MAXYDIM=10.91
      RETURN
      END
```

APPENDIX III. Listing of CDC Fortran IV program TIMLIS.

The first nineteen lines of this appendix are a procedure file which will run program TIMLIS as a batch job on the Cyber. This procedure file will examine the first 1500 records on tape WI20FR and print out the time code of each record. UOILIB contains subroutine GBYTES which is used to convert the IBM 16 bit (positive integer) words to CDC 60 bit (positive integer) words. SYMPLIB contains subroutine SKIPB which is used to backspace one record on the IBM tape.

```
/JOB
/NOSEQ
MKM.
SIGNON(3KEMVUJ)
BILL,ELEC-PS2714.
USE,OPTION,CTIMLST.
GET,OPTION.
PRINT.
GET,CTIMLST.
SETTL,50.
FTN,I=CTIMLST,L=0,A.
LABEL(TAPE,NT,LB=KU,PO=UR,VSN=WI20FR-E598,F=L,D=800)
GRAB,UOILIB.
$ADDLIB,SYMPLIB.
LGO,,,TAPE.
UNLOAD,TAPE.
/EOR
0800
0001,1500,1
```

```
      PROGRAM TIMLIS(INPUT,OUTPUT,IBMTP,TAPE1=INPUT,TAPE2=OUTPUT,
     *               TAPE3=IBMTP)
C
C  THIS PROGRAM IS INTENDED  TO DISPLAY THE HEADER RECORDS AND DATE
C  CODES AND/OR CHECK THE TIME BETWEEN RECORDS ON THE DATA TAPE
C  AGAINST THE EXPECTED VALUE FOR NASA DATA TAPES RECORDED AT WALLOPS
C  ISLAND VIRGINIA.  THE USER SUPPLIES THE BEGINNING RECORD NUMBER
C  AND THE NUMBER OF RECORDS TO BE READ INDICATING IF ALL TIME RECORDS
C  IN THAT BLOCK WILL BE PRINTED OR JUST THOSE WITH TIME LENGTHS NOT
C  EXPECTED.  MORE THAN ONE DATA CARD CAN BE USED.  DATA CARDS MUST
C  HAVE THE FOLLOWING FORMAT:
C
C  FIRST DATA CARD
C         COLUMN 1-4           EXPECTED TIME BETWEEN RECORDS
C                              (MILLISECONDS*10)
C  OTHER DATA CARDS
C         COLUMN 1-4           FIRST RECORD TO BE EXAMINED, RIGHT
                               JUSTIFIED
C         COLUMN 5             COMMA
C         COLUMN 6-9           NUMBER OF CONSECUTIVE RECORDS TO
                               EXAMINE, RIGHT JUSTIFIED
```

```
C         COLUMN 10           COMMA
C         COLUMN 11           1 IF ALL TIME CODES ARE TO BE PRINTED
C                             0 IF ONLY DISCREPANCIES ARE TO BE PRINTED
C  ALL INPUT DATA IS INTEGER FORMAT.
C
C  A 789 CARD MUST FOLLOW THE DATA DECK.  THE PROGRAM WILL PRINT HEADERS
C  FOR RECORDS OF LENGTH 10 OR 45 WORDS.  EACH WORD IS 16 BITS LONG.
C
C  A RECORD CONTAINING A PARITY ERROR WILL BE SKIPPED OVER.
C
C  WRITTEN BY KEITH FRIES    JULY 5, 1978.
C  MODIFIED BY M K MCINERNEY JULY 1979.
C
C  AERONOMY LABORATORY - UNIVERSITY OF ILLINOIS
C
      IMPLICIT INTEGER(A-Z)
      REAL DIFF,DFRSEC,F1
      DIMENSION ARRAY(2008),PARRAY(400)
      DATA (NBLOCK=0)
C
C  THE FOLLOWING LINE LOCATES THE 'FILE ENVIRONMENT TABLE' ADDRESS WHICH
C  IS USED LATER IN THE PROGRAM TO BACKSPACE TAPE3.
C
      NN=FETADR(3)
C
C  SET FLAG FOR PRINTING HEADER WITH FIRST GROUP OF RECORDS.
C
      HEAD=0
C
C  WRITE "TIMLIS" BANNER ON OUTPUT.
C
      WRITE(2,1000)
 1000 FORMAT("0"/"0",6X,"TTTTT  IIIII  M     M  L      IIIII   SSSS ",
     *12X,"AERONOMY"
     */" ",6X,"  T      I    MM   MM  L        I    S    S"
     */" ",6X,"  T      I    M M M M  L        I    S     ",
     *10X,"LABORATORY"
     */" ",6X,"  T      I    M  M  M  L        I     SSSS "
     */" ",6X,"  T      I    M     M  L        I         S",
     *14X,"ROCKET"
     */" ",6X,"  T      I    M     M  L        I    S    S"
     */" ",6X,"  T    IIIII  M     M  LLLLL  IIIII   SSSS ".
     *13X,"PROGRAM"
     */"0",16X,"-- CDC CYBER VERSION --"/"0"/"0")
C
C  READ CARD INFORMATION.
C  INPUT STANDARD TIME BETWEEN RECORDS.
C
      READ(1,1100) STDSTP
 1100 FORMAT(I4)
C
C  INPUT DESIRED RECORD INFORMATION.
C
   10 READ(1,1200) RECNUM,NUM,ALLPRT
```

```
 1200 FORMAT(I4,1X,I4,1X,I1)
      IF(EOF(1)) 210,20
      STOP
C
C  SET FLAG INDICATING FIRST RECORD IN GROUP.
C
   20 START=0
C
C  PRINT INPUT PARAMETERS.
C  START NEW PAGE IF SECOND OR GREATER DATA GROUP.
C
      IF(HEAD.EQ.0) GOTO 30
      WRITE(2,1300)
 1300 FORMAT("1")
   30 HEAD=1
      WRITE(2,1400) NUM,RECNUM,(FLOAT(STDSTP)/10000.0)
 1400 FORMAT("0"/"0 SEARCH ",I4," RECORDS BEGINNING WITH RECORD "
     *,I4,".  EXPECTED TIME BETWEEN RECORDS IS ",F5.4," SECONDS.")
      IF(ALLPRT.EQ.1) GOTO 40
      WRITE(2,1500)
 1500 FORMAT("0 ONLY RECORDS WHICH HAVE NON-STANDARD TIME ",
     *"BETWEEN RECORDS WILL BE PRINTED")
   40 WRITE(2,1600)
 1600 FORMAT("0"/" ")
C
C  SKIP THE PROPER NUMBER OF RECORDS TO GET TO THE RECORDS DESIRED.
C
      ISKIP=RECNUM-NBLOCK-1
      IF(ISKIP) 50,90,70
   50 ISKIP=-ISKIP
      DO 60 I=1,ISKIP
         CALL BACKUP(NN,NBLOCK)
   60 CONTINUE
      GOTO 90
   70 DO 80 I=1,ISKIP
         CALL FORWRD(3,NBLOCK)
   80 CONTINUE
C
C  FILL ARRAY WITH DATA.
C
   90 DO 200 I=1,NUM
         CALL TPGET(3,ARRAY,1,NWORDS,NBLOCK)
         IF(NWORDS.EQ.2008) GOTO 100
         IF(NWORDS.EQ.1005) GOTO 140
         IF(NWORDS.LE.45)   GOTO 150
         WRITE(2,1700) NBLOCK,NWORDS
 1700    FORMAT("+","RECORD ",I4," NON-STANDARD LENGTH, ",I4,
     *   " WORDS."/"0")
         GOTO 200
  100    CALL CALTIM(ARRAY(2006),ARRAY(2007), ARRAY(2008),DAY,HOUR,MIN,
     *   SEC,FRCSEC)
         D1=DAY/100
         D2=(DAY-D1*100)/10
         D3=(DAY-D1*100-D2*10)
```

```
      H1=HOUR/10
      H2=HOUR-H1*10
      M1=MIN/10
      M2=MIN-M1*10
      S1=SEC/10
      S2=SEC-S1*10
      F1=FRCSEC*.0001
C
C  CONVERT TIME TO SECONDS.
C  KEEP IN INTEGER VARIABLES TO AVOID CONVERSION ERROR.
C
      DECSEC=864000000*DAY+36000000*HOUR+600000*MIN+10000*SEC+FRCSEC
      IF(START.EQ.0) GOTO 110
      IDIFF=DECSEC-OLDDEC
C
C  CHECK TIME INTERVAL, IF CORRECT DO NOT PRINT.
C
      IF(IDIFF.EQ.STDSTP) GOTO 110
C
C  BREAK INTO SINGLE DIGIT VARIABLES IN ORDER TO DISPLAY
C  LEADING ZEROS.
C  CONVERT TO REAL TO SIMPLIFY PRINT ROUTINE, CONVERSION ERROR
C  WILL BE ROUNDED VIA PRINT.
C
      DIFF=IDIFF*.0001
      DDAY=DIFF/86400
      DHOUR=(DIFF-DDAY*86400)/3600
      DH1=DHOUR/10
      DH2=DHOUR-DH1*10
      DMIN=(DIFF-DDAY*86400-DHOUR*3600)/60
      DM1=DMIN/10
      DM2=DMIN-DM1*10
      DSEC=DIFF-DDAY*86400-DHOUR*3600-DMIN*60
      DS1=DSEC/10
      DS2=DSEC-DS1*10
      DFRSEC=DIFF-INT(DIFF)
      TEMP=NBLOCK-1
      WRITE(2,1800)NBLOCK,TEMP,DDAY,DH1,DH2,DM1,DM2,DS1,DS2,DFRSEC
 1800 FORMAT("+",60X,"TIME BETWEEN RECORDS ",I4," AND ",I4,
     *  ": DAYS ",I3,", ",2I1,":",2I1,":",2I1,F5.4)
      GOTO 120
  110 IF(ALLPRT.NE.1) GOTO 130
  120 WRITE(2,1900)NBLOCK,D1,D2,D3,H1,H2,M1,M2,S1,S2,F1
 1900 FORMAT("+","RECORD ",I4,"  TIME: DAY ",
     *  3I1,", ",2I1,":",2I1,":",2I1,F5.4," U.T."/"0")
  130 START=1
      OLDDEC=DECSEC
      GOTO 200
  140 WRITE(2,2000) NBLOCK
 2000 FORMAT("+","RECORD ",I4," CONTAINS 1005 WORDS."/"0")
      GOTO 200
  150 IF(NWORDS.EQ.-1) GOTO 190
      WRITE(2,2100) NBLOCK,NWORDS
 2100 FORMAT("+","RECORD ",I4," CONTAINS ",I2," WORDS.")
```

```
C
C  THE FOLLOWING THREE DO-LOOPS ARE NECESSARY TO CONVERT TEXTUAL DATA
C  FROM IBM EBCDIC REPRESENTATION TO CYBER DISPLAY REPRESENTATION.
C
        DO 160 N=1,NWORDS
           M=2*N-1
           CALL GBYTES(ARRAY(N),PARRAY(M),44,8,0,2)
  160   CONTINUE
        M=2*NWORDS
        DO 170 N=1,M
           PARRAY(N)=CONETD(PARRAY(N))
  170   CONTINUE
        DO 180 N=1,M
           IF((PARRAY(N).EQ.00).OR.(PARRAY(N).EQ.42).OR.
     *     (PARRAY(N).EQ.50).OR.(PARRAY(N).EQ.63)) PARRAY(N)=45
  180   CONTINUE
        WRITE(2,2200)(PARRAY(N),N=1,M)
 2200   FORMAT(" ",90R1)
        WRITE(2,2300)
 2300   FORMAT("0")
        GOTO 200
C
C  RESET START FLAG IF PARITY ERROR IS ENCOUNTERED.
C
  190   START=0
  200 CONTINUE
      GOTO 10
  210 STOP
      END
      SUBROUTINE CALTIM(T2006,T2007,T2008,DAY,HOUR,MIN,SEC,FRCSEC)
C
C  SUBROUTINE CALTIM (CALCULATE TIME) CALCULATES TIMES
C  BY DECODING THE NASA DIGITAL TIME DATA IN THE LAST
C  THREE WORDS OF A TAPE RECORD.   THIS IS UNIVERSAL
C  COORDINATED TIME IN DAYS, HOURS, MINUTES, SECONDS,
C  AND DECIMAL FRACTIONAL SECONDS.
C
C  CONVERT T2006 INTO FRACTIONAL SECONDS.
C
      IMPLICIT INTEGER(A-Z)
      DIMENSION I(16)
      CALL BINARY(T2006,I)
      FRCSEC=(8*I(16)+4*I(15)+2*I(14)+I(13))*1000
     *+(8*I(12)+4*I(11)+2*I(10)+I(9))*100
     *+(8*I(8)+4*I(7)+2*I(6)+I(5))*10
     *+(8*I(4)+4*I(3)+2*I(2)+I(1))
C
C  CONVERT T2007 INTO MINUTES AND SECONDS.
C
      CALL BINARY(T2007,I)
      MIN=(4*I(15)+2*I(14)+I(13))*10
     *+(8*I(12)+4*I(11)+2*I(10)+I(9))
      SEC=(4*I(7)+2*I(6)+I(5))*10
     *+(8*I(4)+4*I(3)+2*I(2)+I(1))
```

```
C
C  CONVERT T2008 INTO DAYS AND HOURS.
C
      CALL BINARY(T2008,I)
      DAY=(2*I(16)+I(15))*100
     *+(8*I(14)+4*I(13)+2*I(12)+I(11))*10
     *+(8*I(10)+4*I(9)+2*I(8)+I(7))
      HOUR=(2*I(6)+I(5))*10
     *+(8*I(4)+4*I(3)+2*I(2)+I(1))
      RETURN
      END
      SUBROUTINE BINARY(IINTGR,I)
C
C  SUBROUTINE BINARY GENERATES THE BINARY REPRESENTATION
C  OF THE NUMBER INTGR, FOR USE IN SUBROUTINE CALTIM.
C
C  THE ARRAY I(1-16) IS THE BINARY REPRESENTATION OF
C  THE NUMBER INTGR, WITH I(16) BEING THE MOST SIGNI-
C  FICANT BIT,2**15, AND I(1) THE LEAST SIGNIFICANT
C  BIT,2**0.
C
      INTEGER I(16)
      INTGR=IINTGR
      DO 10 N=1,16
         I(N)=0
   10 CONTINUE
      DO 20 N=1,16
         J=17-N
         L=2**(J-1)
         IF(INTGR.GE.L) I(J)=1
         IF(INTGR.GE.L) INTGR=INTGR-L
   20 CONTINUE
      RETURN
      END
      SUBROUTINE TPGET(U,ARRAY,M,NWORDS,NBLOCK)
C
C  THIS SUBROUTINE IS USED TO TRANSFER ONE RECORD OF TAPE DATA FROM
C  UNIT 'U' TO 'ARRAY', BEGINNING WITH ARRAY ELEMENT M.  THE NUMBER
C  OF 16-BIT WORDS TRANSFERRED ON EACH CALL IS NWORDS (TYPICALLY 10, 45,
C  1005, OR 2008). THE NUMBER OF THE RECORD JUST READ IS NBLOCK.
C
      IMPLICIT INTEGER(A-Z)
      DIMENSION BUFFER(540), ARRAY(1)
      BUFFER IN (U,1)(BUFFER(1),BUFFER(540))
      NBLOCK=NBLOCK+1
      IF(UNIT(U)) 10,20,30
   10 CALL LENGTHX(U,I,J)
C
C  I IS THE NUMBER OF 60-BIT WORDS READ.
C  J IS THE NUMBER OF BITS IN THE LAST 60-BIT WORD THAT WERE NOT USED.
C
      NWORDS=(60*I-J)/16
      CALL GBYTES(BUFFER,ARRAY(M),0,16,0,NWORDS)
      RETURN
```

```
   20 NBLOCK=NBLOCK-1
      WRITE(2,1000) NBLOCK
 1000 FORMAT(" "/"0","*** END OF FILE ENCOUNTERED. LAST RECORD IS ",I4,
     *". ***")
      STOP 1
   30 WRITE(2,1100) NBLOCK
 1100 FORMAT("0*** PARITY ERROR DETECTED IN RECORD ",I4,". ***"/"0"/"0")
      NWORDS=-1
      RETURN
      END
      SUBROUTINE FORWRD(U,NBLOCK)
C
C  THIS SUBROUTINE IS USED TO FORWARD SPACE ONE RECORD ON TAPE UNIT U.
C  THE NUMBER OF THE RECORD SKIPPED IS NBLOCK.
C
      IMPLICIT INTEGER(A-Z)
      DIMENSION BUFFER(540)
      BUFFER IN (U,1)(BUFFER(1),BUFFER(540))
      NBLOCK=NBLOCK+1
      IF(UNIT(U)) 10,20,30
   10 RETURN
   20 NBLOCK=NBLOCK-1
      WRITE(2,1000) NBLOCK
 1000 FORMAT(" "/"0","*** END OF FILE ENCOUNTERED. LAST RECORD IS ",I4,
     *". ***")
      STOP 2
   30 WRITE(2,1100) NBLOCK
 1100 FORMAT("0*** PARITY ERROR DETECTED IN RECORD ",I4,". ***"/"0"/"0")
      RETURN
      END
      INTEGER FUNCTION FETADR(UNIT)
C
C  THIS INTEGER FUNCTION RETURNS AS ITS VALUE THE ADDRESS OF
C  THE "FILE ENVIRONMENT TABLE" CORRESPONDING TO THE FORTRAN
C  UNIT NUMBER GIVEN BY "UNIT", OR ELSE 0, IF THAT UNIT WAS NOT
C  DEFINED IN THE "PROGRAM" STATEMENT IN THE MAIN PROGRAM. FOR
C  INSTANCE, IF THE PROGRAM STATEMENT IS
C
C          PROGRAM XYZ(TAPE1,TAPE2,INPUT,TAPE5=INPUT)
C
C  THEN FETADR(1) WILL BE THE MACHINE ADDRESS OF THE "FET" FOR
C  "TAPE1", FETADR(2) WILL BE ADDRESS OF THE FET FOR TAPE2
C  AND FETADR(5) WILL BE THE ADDRESS OF THE FET FOR FILE "INPUT",
C  SINCE UNIT 5 IS EQUATED TO "INPUT" IN THE PROGRAM. FETADR(6),
C  FOR EXAMPLE, WILL BE 0, SINCE TAPE6 DOES NOT APPEAR IN THE
C  PROGRAM STATEMENT.
C
C  THIS FUNCTION IS INTENDED TO WORK ONLY UNDER CDC CYBER FORTRAN.
C  IF YOU TRY TO USE IT IN ANY OTHER CIRCUMSTANCE, YOU DESERVE
C  WHAT YOU GET. ALSO, UNLESS YOU KNOW WHAT A FET IS AND WHAT IT'S
C  FOR, DON'T MUCK WITH THE ROUTINE, BECAUSE OTHER ROUTINES MAY
C  DEPEND ON THIS ONE.
C
C  CODED FEBRUARY 1978 BY STAN KERR AT THE COMPUTING SERVICES OFFICE
```

```
C  OF THE UNIVERSITY OF ILLINOIS AT URBANA, FOR BOB ZIMMERMAN OF
C  ELECTRICAL ENGINEERING.
C
      IMPLICIT INTEGER(A-Z)
      DIMENSION CORE(1)
      LOF=LOFADR(0)
      IF(UNIT .GE. 10) GOTO 10
         LFN = 4LTAPE + SHIFT(1R0+UNIT,30)
         GOTO 20
   10 LFN = 4LTAPE + SHIFT(1R0+UNIT/10,30) +
     +                SHIFT(1R0+MOD(UNIT,10),24)
   20 B = LOCF(CORE(0))
      FETADR = 0
      DO 30 I = 2,64
         IF (CORE(LOF+I-1-B).EQ. 0) GO TO 50
         IF((CORE(LOF+I-1-B).AND.-777777B).EQ.LFN) GO TO 40
   30 CONTINUE
   40 FETADR=CORE(LOF+I-1-B).AND.777777B
   50 RETURN
      END
          IDENT   LOFADR
          ENTRY   LOFADR
 LOFADR   EQ      0
          SA1     =XLOF.FTN
          BX6     X1
          EQ      LOFADR
          END
      SUBROUTINE BACKUP(FETADR,NBLOCK)
C
C  THIS ROUTINE IS INTENDED MAINLY FOR BACKING UP 1 BLOCK ON
C  A MAGNETIC TAPE BEING READ BY A FORTRAN PROGRAM. IT IS ASSUMED
C  THAT THE TAPE IS ACCESSED BY THE PROGRAM VIA SOME UNIT NUMBER,
C  SAY 5. THE USER OF THIS ROUTINE SHOULD FIRST USE THE INTEGER
C  FUNCTION "FETADR" TO ESTABLISH THE ADDRESS OF THE "FILE
C  ENVIRONMENT TABLE" FOR THE TAPE, AND SAVE THE VALUE (NEVER MIND
C  WHY, JUST DO IT). THEN THIS ROUTINE MAY BE CALLED AT WILL TO
C  BACKSPACE THE TAPE BY ONE BLOCK. THE SECOND PARAMETER IS SUPPOSED
C  TO BE AN INTEGER VARIABLE WHICH THE USER PRESUMABLY USES TO
C  KEEP TRACK OF WHICH BLOCK THE TAPE IS POSITIONED ON; ALL BACKUP
C  DOES WITH "BLOCK" IS SUBTRACT ONE FROM IT.
C
C  IF THE TAPE IS UNIT 5, THEN THE PROCESS OF USING BACKUP WOULD BE
C  SOMETHING LIKE THIS:
C
C      IADDR = FETADR(5)
C
C      .
C      .
C      CALL BACKUP(IADDR,NBLOCK)
C      .
C      .
C      .
C      DO 25 I = 1,10
C 25     CALL BACKUP(IADDR,NBLOCK)
```

```
C
C      ETCETERA,ETCETERA,ETCETERA
C
C  THIS  ROUTINE IS INTENDED TO RUN ONLY UNDER CYBER FORTRAN AT THE
C  UNIVERSITY OF ILLINOIS. IT USES AN EXTERNAL ROUTINE "SKIPB" FROM
C  SYSTEM LIBRARY SYMPLIB TO DO THE ACTUAL BACKSPACE.
C  (TO ADD SYMPLIB TO THE LOCAL LIBRARY ENTER -
C   $ADDLIB,SYMPLIB.)
C
C  CODED BY STAN KERR AT THE COMPUTING SERVICES OFFICE OF THE UNIVERSITY
C  OF ILLINOIS AT URBANA FOR BOB ZIMMERMAN OF ELECTRICAL ENGINEERING.
C
      IMPLICIT INTEGER(A-Z)
      DIMENSION  CORE(1)
      BIAS = LOCF(CORE(0))
      IN = CORE(FETADR+1-BIAS) .AND. 777777B
      CORE(FETADR+2-BIAS) = (CORE(FETADR+2-BIAS) .AND. MASK(42)) + IN
      CORE(FETADR+3-BIAS) = (CORE(FETADR+3-BIAS) .AND. MASK(42)) + IN
      CALL SKIPB(CORE(FETADR-BIAS),1,1)
      NBLOCK = NBLOCK-1
      RETURN
      END
          IDENT  CONETD
((        ENTRY  CONETD
**        CONETD - TABLE ORGANIZATION
*
*         THE CONVERSION TABLE USED BY CONETD IS SET UP WITH
*         8 DISPLAY CODE CHARACTERS PER WORD, LEFT-JUSTIFIED, TO
*         SIMPLIFIY THE ARITHMETIC NECESSARY TO ACCESS A WORD
*         WHICH CONTAINS THE DISPLAY CODE CORRESPONDING TO A GIVEN
*         EBCDIC CHARACTER.
*
CONETD    EQ     0
          SA1    X1           X1 = I
          SX6    -1           X6=-1, DEFAULT VALUE IF I OUT OF RANGE
          NG     X1,CONETD    RETURN IF I<0
          SX2    256
          IX2    X1-X2
          PL     X2,CONETD    RETURN IF I>255
          BX2    X1
          AX2    3            X2=I/8
          SA2    X2+TABLE     X2 = TABLE WORD WITH CONVERTED VALUE
          MX0    57
          BX1    -X0*X1       X1=MOD(I,8)
          SX1    X1+1         X1=MOD(I,8)+1
          LX1    1
          BX3    X1
          LX3    1
          IX1    X1+X3        X1=6*(1+MOD(I,8))
          SB2    X1
          LX2    B2           SHIFT CONVERTED VALUE TO LOW 6 BITS
          MX0    54
          BX6    -X0*X2       AND OUT EXTRANEOUS AND RETURN
          EQ     CONETD
```

```
*
TABLE     DATA    55626460555255650000B,55555455564657500000B
          DATA    33343536555551550000B,43445555615473710000B
          DATA    55555555554742770000B,55555555550067700000B
          DATA    55554155555555530000B,55555555374055000000B
          DATA    55555555555555550000B,55556157725145660000B
          DATA    67555555555555550000B,55556253475277760000B
          DATA    46505555555555550000B,55557556636573710000B
          DATA    55555555555555550000B,55740060747054640000B
          DATA    55010203040506070000B,10115555555555550000B
          DATA    55121314151617200000B,21225555555555550000B
          DATA    55762324252627300000B,31325555555555550000B
          DATA    55555555555555550000B,55555555555555550000B
          DATA    72010203040506070000B,10115555555555550000B
          DATA    66121314151617200000B,21225555555555550000B
          DATA    75552324252627300000B,31325555555555550000B
          DATA    33343536374041420000B,43445555555555550000B
          END
```

APPENDIX IV. Listing of IBM Fortran IV program FFTR.

The JCL and input parameters necessary for processing Faraday rotation data from flight 18.1020 are included. The Fourier transform is computed by subroutine FFTRC, and IMSL library routine.

The data lines following the program result in the processing of 34 seconds (T+50 → T+83) for channel 4.

```
//MIKE1 JOB
/*ID PS=2714,NAME='MCINERNEY'
/*ID CODE=
/*ID BIN=49
/*ID EJECT=YES
/*ID PLOT=YES,IOREQ=5000,LINES=5500,TIME=(3,00),REGION=280K
/*SETUP UNIT=TAPE,R=E598,ID=(WI20FR,NORING)
// EXEC FORTLDPZ,PARM.PLOT='TIME=15,LENGTH=60',REGION.GO=280K
//FORT.SYSIN DD *
C
C                     ----- PROGRAM FFTR -----
C
C  PROGRAM FOR COMPUTING THE FOURIER TRANSFORM OF A SIGNAL, CALCULATING
C  THE FREQUENCY OF THE SPACTRAL PEAKS AND PLOTTING THE SPECTRUM.
C
C  A GAUSSIAN WINDOW IS APPLIED TO THE DATA BEFORE BEING TRANSFORMED.
C
C  INPUT PARAMETERS:
C
C    1: FRSTRC IS THE NUMBER OF THE RECORD WHERE THE FIRST DATA
C                POINT IS LOCATED.
C    2: FRSTFM IS THE NUMBER OF THE FRAME IN FRSTRC WHERE THE
C                FIRST DATA POINT IS LOCATED.
C    3: CHANL IS THE NUMBER OF THE CHANNEL TO BE PROCESSSED.
C    4: LENGTH IS THE NUMBER OF POINTS TO BE FOURIER TRANSFORMED.
C
C  ALL PARAMETERS ARE ENTERED ON A SINGLE LINE WITH THE FOLLOWING
C  FORMAT:
C
C        I5,5 SPACES,I4,SPACE,I1,4 SPACES,I5.
C
C  EXAMPLE:
C         00750     0128 3    05000
C
C     COLUMN 1
C
C  THIS EXAMPLE INDICATES THE PROCESSING OF 5000 CONSECUTIVE DATA
C  POINTS FROM CHANNEL 3.  THE FIRST POINT TO BE TAKEN FROM
C  RECORD 750, FRAME 128.
C
C  WARNING - USER MUST  COMPUTE ALL OF THE ABOVE VALUES CAREFULLY.
C
C  NOTE:
C  IWK MUST BE FOUND IN CORE ON A BOUNDRY DIVISIBLE BY 8.
C  INTEGER*4 IWK(18050) WILL ONLY WORK HALF THE TIME.
```

```
C  THIS IS BECAUSE IT IS USED AS A STORAGE AREA FOR A REAL *4
C  VARIABLE IN FFTRC.
C
C  MODIFIED BY M K MCINERNEY SEPTEMBER 1979.
C
      INTEGER*2 ARRAY(2010)
      INTEGER CHANL,EOF,ERRAD,ERRCD,FRSTFM,FRSTPT,FRSTRC,LENGTH,TPPROC
      REAL*4 DMAG(3001),FREQ(200),SCA(2),SCF(2)
      REAL*8 DATA(6002),IWK(9025),WK(9025)
      EQUIVALENCE(DATA,DMAG)
      EQUIVALENCE(IWK,WK)
      ASSIGN 250 TO ERRAD
      ASSIGN 260 TO EOF
      WRITE (6,1000)
 1000 FORMAT('1'
     */'0',6X,'FFFFF  FFFFF  TTTTT  RRRRR ',12X,'AERONOMY'
     */' ',6X,'F      F        T    R    R'
     */' ',6X,'F      F        T    R    R',10X,'LABORATORY'
     */' ',6X,'FFFF   FFFF     T    RRRRR '
     */' ',6X,'F      F        T    R  R  ',14X,'ROCKET'
     */' ',6X,'F      F        T    R   R '
     */' ',6X,'F      F        T    R    R',13X,'PROGRAM'/'0')
C
C  GENERATE FREQUENCY ARRAY (MINUS TO MOVE DOWN THE PAGE).
C
      DO 10 NF=1,200
         FREQ(NF)=FLOAT(1-NF)
   10 CONTINUE
C
C  INITIALIZE PLOT VARIABLES, COUNTERS AND OPEN TAPE FILE.
C
      SCF(1)=-180.0
      SCF(2)=20.0
      NCRV=0
      NBLOCK=0
      CALL PLOTS(0.,0.,99)
      CALL PLOT(.5,.5,-3)
      CALL TPOPIZ(12,ERRAD,ERRCD)
C
C  READ PARAMETERS.
C
   20 READ (5,1100,END=240) FRSTRC,FRSTFM,CHANL,LENGTH
 1100 FORMAT(I5,5X,I4,1X,I1,4X,I5)
      NCRV=NCRV+1
C
C  POSITION TAPE AT STARTING RECORD.
C
   30 NSKIP=FRSTRC-NBLOCK-1
      IF(NSKIP) 40,80,60
   40 NSKIP=-NSKIP
      TPPROC=-1
      DO 50 I=1,NSKIP
         CALL TPBSRZ(12)
         NBLOCK=NBLOCK-1
```

```
   50 CONTINUE
      GOTO 80
   60 TPPROC=1
      DO 70 I=1,NSKIP
         CALL TPFSRZ(12)
         NBLOCK=NBLOCK+1
   70 CONTINUE
C
C  WRITE OUT PARAMETERS FOR DIAGNOSTICS.
C
   80 WRITE (6,1200)
 1200 FORMAT(//' ',120('-')//)
      WRITE (6,1300) FRSTRC,FRSTFM,CHANL,LENGTH
 1300 FORMAT('OSTARTING RECORD NUMBER = ',I5/11X,'FRAME NUMBER = ',
     *I5/9X,'CHANNEL NUMBER = ',I5/7X,'NUMBER OF POINTS = ',I5)
C
C  FILL ARRAY DATA WITH 'LENGTH' NUMBER OF DATA POINTS.
C
      INDEX=1
      FRSTPT=(FRSTFM*5)+CHANL+5
      TPPROC=0
   90 CALL TPGETZ(12,ARRAY(1))
      CALL TPCHKZ(12,NBYTES,EOF)
      NBLOCK=NBLOCK+1
      DO 100 I=FRSTPT,2005,5
         DATA(INDEX)=ARRAY(I)
         INDEX=INDEX+1
         IF(INDEX.GT.LENGTH) GOTO 110
  100 CONTINUE
      FRSTPT=CHANL+5
      GOTO 90
C
C  REMOVE AVERAGE DATA VALUE (DC COMPONENT).
C
  110 TOTAL=0.0
      DO 120 N=1,LENGTH
         TOTAL=TOTAL+DATA(N)
  120 CONTINUE
      TOTAL=TOTAL/FLOAT(LENGTH)
      DO 130 N=1,LENGTH
         DATA(N)=DATA(N)-TOTAL
  130 CONTINUE
C
C  APPLY A GAUSSIAN WINDOW TO DATA.
C
      NELN2=LENGTH/2
      FLN2=NELN2
      STT=1.+1./FLOAT(LENGTH)
      GBASE=0.01111
      DO 140 K=1,NELN2
         DEL=3.0*(STT-FLOAT(K)/FLN2)
         G=EXP(-0.5*DEL*DEL)-GBASE
         DATA(K) = G*DATA(K)
         KTP=LENGTH-K+1
```

```
         DATA(KTP) = G*DATA(KTP)
  140 CONTINUE
C
C  TAKE FAST FOURIER TRANSFORM.
C
      CALL FFTRC (DATA,LENGTH,DATA,IWK,WK)
C
C  ARRAY 'DATA' NOW CONTAINS THE COMPLEX SPECTRUM.
C
C  CONVERT TO MAGNITUDES.
C
      DO 150 N=1,NELN2
         NT=N*2
         DMAG(N)=DSQRT(DATA(NT)*DATA(NT)+DATA(NT-1)*DATA(NT-1))
  150 CONTINUE
C
C  WRITE OUT MAGNITUDES.
C
C  WRITE OUT LOWER FREQUENCIES (0-69).
C
      DO 160 N=1,70,10
         NFREQ=N-1
         NTP=N+9
         WRITE (6,1400) NFREQ,(DMAG(K),K=N,NTP)
 1400    FORMAT(1X,I9,10(1X,E9.2))
  160 CONTINUE
C
C  WRITE OUT HIGH FREQUENCIES (400-559).
C
      DO 170 N=401,560,10
         NFREQ=N-1
         NTP=N+9
         WRITE (6,1400) NFREQ,(DMAG(K),K=N,NTP)
  170 CONTINUE
C
C  FIND THE FREQUENCY OF THE MAXIMUM PEAK BETWEEN 3 AND 40 HZ.
C
      K1=4
      K2=41
      MAX=5
      AMAX=0.
      DO 180 K=K1,K2
         IF(DMAG(K).LT.AMAX) GOTO 180
         MAX=K
         AMAX=DMAG(K)
  180 CONTINUE
C
C  COMPUTE FREQUENCY OF SIGNAL ALGORITHM.
C
      MAXM=MAX-1
      MAXP=MAX+1
      RR=ALOG(AMAX/DMAG(MAXP))/ALOG(AMAX/DMAG(MAXM))
      FMAX1=MAX-1
      FSIG=FMAX1+0.5*(1.0-RR)/(1.0+RR)
```

```
      WRITE (6,1500) FSIG
 1500 FORMAT(/,' FREQ. OF PEAK=',F10.5)
C
C  FIND ALL PEAKS BETWEEN 0 AND 69 HZ AND BETWEEN 470 AND 558 HZ.
C
      K1=5
      K2=70
      XKM1=DMAG(1)
      XKM=DMAG(2)
      XK=DMAG(3)
      XKP=DMAG(4)
      GOTO 200
  190 K1=475
      K2=559
      XKM1=DMAG(471)
      XKM=DMAG(472)
      XK=DMAG(473)
      XKP=DMAG(474)
C
C  A SPECTRAL PEAK IS INDICATED BY
C  DMAG(F+1) < DMAG(F) > DMAG(F-1) > DMAG(F-2).
C
  200 DO 230 K=K1,K2
         IF(XK.GT.XKM.AND.XKM.GT.XKM1.AND.XK.GT.XKP) GOTO 220
  210    XKM1=XKM
         XKM=XK
         XK=XKP
         XKP=DMAG(K)
         GOTO 230
C
C  WARNING:  DATA CAN RESULT IN DIVISION BY ZERO.
C
  220    MAXM=K-3
         RR=ALOG(XK/XKP)/ALOG(XK/XKM)
         FSIG=MAXM+0.5*(1.-RR)/(1.+RR)
         WRITE (6,1500) FSIG
         GOTO 210
  230 CONTINUE
      IF(K1.LT.200) GOTO 190
C
C  ***************     PLOTTING SEQUENCE     ***************
C                        SPECTRAL PEAKS
C
C  SET UP SCALING FOR ALL PLOTS.
C   WARNING **  XLOGZ EXPECTS REF. PT. IN LOWER LEFT.
C   WARNING **  DMAG SCALE HAS BEEN SET TO "NICE" VALUES.
C   WARNING **  MINIMUM ALLOWED VALUE OF "DIST" IS 2.0.
C  SCA WILL NOW BE SET TO FIXED VALUES.
C
      SCA(1)=1.
      IF(CHANL.EQ.5) SCA(1)=0.0
      SCA(2)=2.
      CALL XLOGZ (DMAG,FREQ,40,1,SCA,SCF)
      CALL XLOGZ (DMAG(421),FREQ(41),139,1,SCA,SCF)
```

```
      CALL PLOT (1.,0.,-3)
C
C  RETURN TO READ NEXT DATA CARD.
C
      GOTO 20
C
C                            DRAW AXIS
C
C  DRAW FREQUENCY AXIS.
C  START AT ZERO, THEN GO AT 20. PER INCH(FOR 8 INCH AXIS).
C  NCRV IS THE NUMBER OF SPECTRAL PLOTS MADE.
C
  240 DIST=FLOAT(NCRV)*(-1.0)
      SCF(1)=0.0
      SCF(2)=20.
      CALL PLOT (DIST,0.,-3)
      CALL CCP5AX (0.,9.,'LOW FREQ (HZ)',-13,2.,-90.,SCF)
      SCF(1)=420
      CALL CCP5AX (0.,7.,'FREQUENCY (HZ)',-14,7.,-90.,SCF)
C
C  DRAW TIME AXIS.
C
      SCA(1)=-1.
      SCA(2)=1.
      DIST=(DIST*(-1.0))+1.0
      CALL CCP5AX (0.,0.,'TIME(RELATIVE)',-14,DIST,0.,SCA)
      CALL CCP5AX (0.,9.,'TIME(RELATIVE)',14,DIST,0.,SCA)
C
C  **************     END PLOTTING SEQUENCE     ***************
C
      CALL PLOT(0.,0.,999)
      STOP
C
C  COME HERE FOR A PARITY ERROR WHILE READING A RECORD.
C  DISCONTINUE SPECTRAL PROCESSING OF THIS DATA SET
C  AND CONTINUE ON TO THE NEXT ONE.
C
  250 NBLOCK=NBLOCK+1
      WRITE (6,1600) NBLOCK,ERRCD
 1600 FORMAT('0   ERROR IN READING RECORD ',I5,'.  TAPE ERROR ',I10,'.'/
     *'0   SPECTRAL PROCESSING OF THIS DATA SET ABANDONED.'/)
      CALL PLOT(1.,0.,-3)
      IF(TPPROC) 30,20,30
C
C  COME HERE FOR AN END OF FILE ON TAPE.
C
  260 WRITE(6,1700) NBLOCK
 1700 FORMAT('0 END OF FILE ON TAPE.'/'0 LAST RECORD READ WAS ',I5,'.')
      CALL PLOT(0.,0.,999)
      STOP
      END
//GO.FT12F001 DD UNIT=TAPE,VOL=(,RETAIN,SER=WI20FR),DISP=OLD,
// DCB=(RECFM=U,BLKSIZE=4016),LABEL=(1,BLP,,IN)
//GO.SYSIN DD *
```

00995	0090 4	05000
01007	0290 4	05000
01020	0090 4	05000
01032	0290 4	05000
01045	0090 4	05000
01057	0290 4	05000
01070	0090 4	05000
01082	0290 4	05000
01095	0090 4	05000
01107	0290 4	05000
01120	0090 4	05000
01132	0290 4	05000
01145	0090 4	05000
01157	0290 4	05000
01170	0090 4	05000
01182	0290 4	05000
01195	0090 4	05000
01207	0290 4	05000
01220	0090 4	05000
01232	0290 4	05000
01245	0090 4	05000
01257	0290 4	05000
01270	0090 4	05000
01282	0290 4	05000
01295	0090 4	05000
01307	0290 4	05000
01320	0090 4	05000
01332	0290 4	05000
01345	0090 4	05000
01357	0290 4	05000
01370	0090 4	05000
01382	0290 4	05000
01395	0090 4	05000
01407	0290 4	05000

/*

APPENDIX V. Listing of CDC Fortran IV program FR2NE.

The first forty lines of this appendix are Cyber Operating System commands which run program FR2NE as a batch job. The lines following the /EOR statement are input data for program FR2NE. Twenty-one Faraday rates from the 18.1020 5.040 MHz propagation experiment are input. In this example subroutine PRESSR and the three subroutines listed after PRESSR (MONTH, INTER and ROUND) in this appendix are pre-compiled and stored in a user library named BLIB. The four lines following the LGO statement append the file BODY to the file HEADER and remove the file separator (/EOS).

```
/JOB
/NOSEQ
MKM.
SIGNON(3MIKEKM)
BILL,ELEC,PS2714.
USE,OPTION,FR2NE,BLIB.
GET,OPTION.
PRINT.
GET,FR2NE,BLIB.
ADDLIB,BLIB.
R.FTN,I=FR2NE,L=0,A.
LGO,,BODY,HEADER.
REWIND,HEADER,BODY.
SKIPR,HEADER.
COPYEI,BODY,HEADER.
PACK,HEADER.
PRINT/NORIGHT,HEADER.
/EOR
18.1020    1979.15068   5040000.   630000.
   32.8400    73.0496     48207.  1293.0000    50.9961   -93.4361    -3.8052
   32.2826    73.1569     49458.  1285.0000    50.9989   -93.4357    -7.5078
   31.7055    73.2503     50696.  1273.0000    51.0018   -93.4354     1.4364
   31.1363    73.3381     51923.  1262.0000    51.0046   -93.4351     0.8784
   30.6291    73.4249     53140.  1247.0000    51.0074   -93.4348     1.6380
   30.2047    73.5037     54345.  1238.0000    51.0101   -93.4343    75.0960
   29.8226    73.5658     55541.  1231.0000    51.0128   -93.4335    -1.2906
   29.4180    73.6137     56727.  1224.0000    51.0156   -93.4329    -4.1760
   28.9623    73.6549     57901.  1213.0000    51.0186   -93.4325    -2.0646
   28.4880    73.6930     59065.  1203.0000    51.0215   -93.4322    -1.6434
   28.0555    73.7311     60213.  1185.0000    51.0244   -93.4319     0.8478
   27.6927    73.7730     61357.  1182.0000    51.0271   -93.4314     1.3284
   27.3744    73.8130     62505.  1190.0000    51.0298   -93.4308     1.2366
   27.0603    73.8427     63652.  1185.0000    51.0325   -93.4302     1.5462
   26.7380    73.8658     64783.  1160.0000    51.0353   -93.4296     1.4418
   26.4214    73.8904     65893.  1147.0000    51.0380   -93.4291    12.9456
   26.1256    73.9154     66998.  1143.0000    51.0407   -93.4285    12.7350
   25.8515    73.9350     68096.  1134.0000    51.0434   -93.4279    10.1196
   25.5850    73.9479     69182.  1122.0000    51.0462   -93.4273    25.3242
   25.3063    73.9554     70253.  1111.0000    51.0489   -93.4267    76.0572
   25.0053    73.9596     71316.  1106.0000    51.0517   -93.4263    -7.0722
```

```
      PROGRAM FR2NE(INPUT,OUTPUT,HEADNG,TAPE1=INPUT,TAPE2=OUTPUT,
     *              TAPE3=HEADNG)
C
C  ANALYSIS OF FARADAY ROTATION WITHOUT DIFFERENTIAL ABSORPTION
C  GIVEN A MODEL FOR THE COLLISION FREQUENCY.
C
C  THE FIRST INPUT CARD CONTAINS FLIGHT INFORMATION.
C
C          COLUMNS  1-10         ROCKET NUMBER
C          COLUMNS 11-20         DATE (IN DECIMAL PARTS OF YEARS
C                                   E.G. FEBRUARY 26,1979 = 1979.15616)
C          COLUMNS 21-30         FREQUENCY (HERTZ)
C          COLUMNS 31-40         COLLISION FREQUENCY MODEL
C                                PARAMETER (HERTZ/PASCAL)
C          FORMAT(F10.4,F10.5,F10.0,F10.0)
C
C  THE REMAINING DATA CARDS CONTAIN THE ROCKET LOCATION AND
C  VELOCITY ALONG WITH THE EXPERIMENTAL FARADAY RATE.
C
C          COLUMNS  1-10         AZIMUTH (DEGREES)
C          COLUMNS 11-20         ELEVATION (DEGREES)
C          COLUMNS 21-30         HEIGHT (METERS)
C          COLUMNS 31-40         TOTAL VELOCITY (METERS/SECOND)
C          COLUMNS 41-50         ROCKET LATITUDE (DEGREES)
C          COLUMNS 51-60         ROCKET LONGITUDE (DEGREES)
C          COLUMNS 61-70         FARADAY ROTATION (EXPERIMENTAL VALUE)
C                                (DEGREES/SECOND)
C          FORMAT(2F10.4,F10.0,4F10.4)
C
C                       **** IMPORTANT NOTE ****
C  THE PRESSURE IS COMPUTED IN A USER LIBRARY SUBROUTINE NAMED
C  PRESSR.  MAKE SURE THAT YOU HAVE EITHER CALLED THE LIBRARY OR
C  HAVE MADE IT A LOCAL SUBROUTINE BY INSERTING IT AT THE
C  END OF THIS PROGRAM.
C
C  MODIFIED BY M K MCINERNEY OCTOBER 1979.
C
C  COMMON VARIABLES TO BE USED IN FARADAY RATE SUBPROGRAM,
C  MFR, AND COMMON VARIABLES TO BE USED IN SUBROUTINE
C  NPRNT, WHICH PRINTS OUT THE VARIABLE VALUES.
C
      REAL CF,ED,ED1,ED2,MFR,S,W,X,Z
      COMMON/AREA1/S,FC
      COMMON/AREA2/ED,X,Z,AZD,BN,F,HT,ELD,BD,RLATD,BW,RLNGD,B,DAE,
     *V,Y,TH,DA,AC,FR,FRE
      COMMON/AREA3/W,CF,ROR,RIO,RXR,RIX,ROI,AIO,RXI,AIX,CTH
      COMMON/COEFFS/GAUSS(11,11)
C
C  ENTER PARAMETERS OF OBLATE EARTH.
C
      A=6.378165E6
      A2=A*A
      A4=A2*A2
```

```
      FLAT=1.-1./298.3
      B2=(A*FLAT)**2
      A2B2=A2*(1.-FLAT**2)
      A4B4=A4*(1.-FLAT**4)
C
C  ENTER FLIGHT PARAMETERS.
C
      READ(1,1000) RKTNO,TM,F,CFM
 1000 FORMAT(F10.4,F10.5,F10.0,F10.0)
      WRITE(3,1100)
 1100 FORMAT("1INPUT DATA DECK:")
      WRITE(3,1200) RKTNO,TM,F,CFM
 1200 FORMAT("-ROCKET NUMBER     DATE      FREQUENCY      COLLISION FREQUE
     *NCY MODEL PARAMETER"/"0    ",F7.4,5X,F10.5,1X,F10.0,17X,F10.0/"0")
      WRITE(3,1300)
 1300 FORMAT(3X,"AZIMUTH   ELEVATION   ALTITUDE  VELOCITY  LATITUDE   LO
     *NGITUDE        EXPERIMENTAL FARADAY RATE        PRESSURE"/"0")
C
C  INITIALIZE THE COEFFICIENTS USED IN CALCULATING THE
C  MAGNETIC FIELD, COMPUTE THE RADIAN FREQUENCY AND
C  DEFINE A DEGREES-TO-RADIANS CONVERSION VARIABLE.
C
      CALL COEFF(TM)
      W=6.2831853*F
      DTR=1.745329E-2
C
C  ENTER VARIABLES..
C
   10 READ(1,1400) AZD,ELD,HT,V,RLATD,RLNGD,FRE
 1400 FORMAT(2F10.4,F10.0,4F10.4)
      IF(EOF(1).NE.0.0) GOTO 120
      CALL PRESSR (RLATD,HT,TM,P)
      WRITE(3,1500) AZD,ELD,HT,V,RLATD,RLNGD,FRE,P
 1500 FORMAT(2X,F9.4,2X,F9.4,4X,F7.0,4X,F5.0,3X,F8.4,3X,F9.4,16X,F10.4,
     *13X,F10.5/" ")
      DAE=0.0
      CF=CFM*P
C
C  CONVERT DEGREES TO RADIANS.
C
      AZ=AZD*DTR
      EL=ELD*DTR
      RLATR=RLATD*DTR
      RLNGR=RLNGD*DTR
C
C  FIND GEOCENTRIC COORDINATES OF ROCKET.
C
      SINLA=SIN(RLATR)
      SINLA2=SINLA*SINLA
      COSLA2=1.-SINLA2
      DEN2=A2-A2B2*SINLA2
      DEN=SQRT(DEN2)
      FAC=(((HT*DEN)+A2)/((HT*DEN)+B2))**2
      CT=SINLA/SQRT(FAC*COSLA2+SINLA2)
```

```
      R=SQRT(HT*(HT+2.*DEN)+(A4-A4B4*SINLA2)/DEN2)
      ST=SQRT(1.-CT**2)
C
C  CALCULATE GEOMAGNETIC FIELD AT ROCKET.
C
      SPH=SIN(RLNGR)
      CPH=COS(RLNGR)
      CALL FIELD (R,ST,CT,SPH,CPH,BR,BT,BP,B)
C
C  TRANSFORM FIELD COMPONENTS, GEOCENTRIC TO GEODETIC.
C
      SIND=SINLA*ST-SQRT(COSLA2)*CT
      COSD=SQRT(1.-SIND**2)
      BN=-BT*COSD-BR*SIND
      BD=BT*SIND-BR*COSD
      BW=-BP
      S=-1.758803D11*B
      Y=S/W
C
C  CALCULATE COSINE OF PROPAGATION ANGLE AND PROPAGATION ANGLE.
C
      CEL=COS(EL)
      CTH=(CEL*COS(AZ)*BN-SIN(EL)*BD-CEL*SIN(AZ)*BW)/B
      TH=ACOS(CTH)/DTR
C
C  CALCULATE FR AND DA COEFFICIENTS.
C
      FV=F*V
      FC=6.004153E-7*FV
      AC=1.820428E-7*FV
C
C  ITERATE TO MATCH FRE.
C
      WRITE(2,1600) RKTNO,TM,CFM
 1600 FORMAT("1  ",F10.4,F10.3,1PE10.2)
C
C  FOR AN INITIAL GUESS AS TO THE ELECTRON DENSITY WHICH CORRESPONDS
C  TO THE EXPERIMENTAL FARADAY ROTATION RATE, LOCATE IN WHICH ELECTRON
C  DENSITY DECADE BETWEEN 1.0E00 AND 1.0E20 THE EXPERIMENTAL FARADAY
C  ROTATION RATE FALLS.
C
C  IF THE THEORETICAL FARADAY ROTATION RATE IS DISCONTINUOUS WITHIN
C  THE DECADE SEPARATE THE ANALYSIS (CHECK) BASED UPON THE TYPE OF
C  DISCONTINUITY.
C
C     FOR A NEGATIVE-GOING DISCONTINUITY, CHECK TO SEE IF THE EXP FR
C     RATE IS GREATER THAN (OR EQUAL TO) THE MINIMUM NEGATIVE
C     THEORETICAL FR RATE.
C     FOR A POSITIVE-GOING DISCONTINUITY, CHECK TO SEE IF THE EXP FR
C     RATE IS LESS THAN (OR EQUAL TO) THE MAXIMUM POSITIVE
C     THEORETICAL FR RATE.
C
      ED1=1.0E+00
      FR1=MFR(ED1)
```

```
   20 ED2=10.0*ED1
      IF(ED2.GT.1.0E+20) GOTO 110
      FR2=MFR(ED2)
      IF((FR1*FR2).LE.0.0) GOTO 30
      IF(((FRE-FR2)*(FRE-FR1)).LE.0.0) GOTO 50
      GOTO 40
   30 IF(FR1.GT.FR2.AND.FRE.GE.FR2) GOTO 50
      IF(FR1.LT.FR2.AND.FRE.LE.FR2) GOTO 50
   40 ED1=ED2
      FR1=FR2
      GOTO 20
C
C  ITERATE BY HALVING INTERVAL TO MATCH THE EXPERIMENTAL FARADAY RATE.
C
C  IF THE THEORETICAL FR INTERVAL IS DISCONTINUOUS THEN SEPARATE THE
C  ANALYSIS INTO TWO CASES.  ANALYZE BASED UPON WHICH THEORETICAL FR
C  RATE END VALUE HAS THE SAME SIGN AS THE THEORETICAL FR RATE WHICH
C  CORRESPONDS TO THE CENTER ED VALUE.
C
   50 DO 100 I=1,40
         ED=(ED1+ED2)/2.0
         FR=MFR(ED)
         IF(MOD(I,2).EQ.0) CALL NPRNT
         IF((FR1*FR2).LE.0.0) GOTO 70
   60    IF(((FRE-FR)*(FRE-FR1)).LE.0.0) GOTO 90
         GOTO 80
   70    IF((FR1*FR).GT.0.0) GOTO 60
         IF(((FRE-FR)*(FRE-FR2)).GT.0.0) GOTO 90
   80    ED1=ED
         FR1=FR
         GOTO 100
   90    ED2=ED
         FR2=FR
  100 CONTINUE
      GOTO 10
C
C  PRINT OUT ERROR DIAGNOSTIC IF EXPERIMENTAL FARADAY ROTATION
C  VALUE WAS UNCOMPARABLE.
C
  110 WRITE(2,1700) FRE
 1700 FORMAT("OELECTRON DENSITIES BETWEEN 1.0E00 AND 1.0E20 HAVE BEEN SC
     *ANNED."/"ONO DECADE WAS FOUND WHICH CONTAINED A FARADAY ROTATION V
     *ALUE"/"OOF ",F10.5,"."/"OTHIS FARADAY ROTATION VALUE IS PROBABLY U
     *NMATCHABLE.")
      GOTO 10
  120 WRITE(3,1300)
      WRITE(3,1800)
      WRITE(3,1900)
 1800 FORMAT("1NOTE THE FOLLOWING ABREVIATIONS AND CONVENTIONS:"/
     *"   AZ  - AZIMUTH OF ROCKET FROM TRANSMITTER - - - - DEGREES"/
     *"   EL  - ELEVATION OF ROCKET FROM TRANSMITTER  - - - DEGREES"/
     *"   LAT - GEODETIC LATITUDE OF ROCKET  - - - - - - - DEGREES"/
     *"   LNG - ROCKET LONGITUDE - - - - - - - - - - - - - DEGREES"/
     *"   V   - TOTAL VELOCITY - - - - - - - - - - - METERS/SECOND"/
```

```
     *"   BN  - GEOMAGNETIC FLUX (NORTH COMPONENT)  - - - -   TESLA"/
     *"   BD  - GEOMAGNETIC FLUX (DOWNWARD COMPONENT)  - - -   TESLA"/
     *"   BW  - GEOMAGNETIC FLUX (WESTWARD COMPONENT)  - - -   TESLA"/
     *"   B   - TOTAL GEOMAGNETIC FLUX  - - - - - - - - -   TESLA"/
     *"   Y   - RATIO OF GYRO TO WAVE FREQUENCY  - - - - - - - -  "/
     *"   ROR - POLARIZATION (ORDINARY)(REAL PART)  - - - - - - -  "/
     *"   RXR - POLARIZARION (EXTRAORDINARY)(REAL PART)  - - - -  "/
     *"   ROI - POLARIZATION (ORDINARY)(IMAGINARY PART)  - - - -  "/
     *"   RXI - POLARIZATION (EXTRAORDINARY)(IMAGINARY PART) - - -  "/
     *"   CTH - COSINE OF TH  - - - - - - - - - - - - -  ")
 1900 FORMAT(
     *"   RIO - REFRACTIVE INDEX (ORDINARY).  - - - - - - - - -  "/
     *"   RIX - REFRACTIVE INDEX (EXTRAORDINARY) - - - - - - - -  "/
     *"   AIO - ABSORPTION INDEX (ORDINARY)  - - - - - - - - -  "/
     *"   AIX - ABSORPTION INDEX (EXTRAORDINARY) - - - - - - - -  "/
     *"   TH  - ANGLE BETWEEN WAVE NORMAL & GEOMAGNETIC FIELD  DEGREES"/
     *"   F   - FREQUENCY  - - - - - - - - - - - - -  HERTZ"/
     *"   FRE - FARADAY RATE, EXPERIMENTAL - - - -  DEGREES/SECOND"/
     *"   FR  - FARADAY RATE, COMPUTATIONAL  - - -  DEGREES/SECOND"/
     *"   DAE - DIFFERENTIAL ABSORBTION RATE, EXPERIMENTAL  DBS/SECOND"/
     *"   DA  - DIFFERENTIAL ABSORPTION RATE, COMPUTATIONAL DBS/SECOND"/
     *"   HT  - ALTITUDE - - - - - - - - - - - - - -  METERS"/
     *"   ED  - ELECTRON DENSITY  - - - - - - - - -  METERS**-3"/
     *"   X   - RATIO OF PLASMA TO WAVE FREQUENCY  - - - - - -  "/
     *"   CF  - COLLISION FREQUENCY  - - - - - - - - -  HERTZ"/
     *"   Z   - RATIO OF COLLISION FREQUENCY TO ANGULAR WAVE FREQUENCY")
      STOP
      END
      REAL FUNCTION MFR(ED)
C
C  THIS FUNCTION COMPUTES THE FARADAY RATE AS A FUNCTION
C  OF ELECTRON DENSITY.
C
      REAL AIO,AIX,CF,CTH,ED,FC,RIO,RIX,ROI,ROR,RXI,RXR,S,W
      COMMON/AREA1/S,FC
      COMMON/AREA3/W,CF,ROR,RIO,RXR,RIX,ROI,AIO,RXI,AIX,CTH
      CALL SENWYL (ED,CF,S,CTH,W,RIO,RIX,AIO,AIX,ROR,RXR,ROI,RXI)
      MFR=FC*(RIO-RIX)
      RETURN
      END
      SUBROUTINE NPRNT
C
C  THIS SUBROUTINE PRINTS OUT THE VALUES OF THE VARIABLES USED IN
C  FINDING THE ELECTRON DENSITY.
C
      REAL AC,AIO,AIX,AZD,B,BD,BN,BW,CF,CTH,DA,DAE,ED,ELD,F,FR,FRE,
     *HT,RIO,RIX,RLATD,RLNGD,ROI,ROR,RXI,RXR,TH,V,W,X,Y,Z
      COMMON/AREA2/ED,X,Z,AZD,BN,F,HT,ELD,BD,RLATD,BW,RLNGD,B,DAE,
     *V,Y,TH,DA,AC,FR,FRE
      COMMON/AREA3/W,CF,ROR,RIO,RXR,RIX,ROI,AIO,RXI,AIX,CTH
C
C  CALCULATE ABSORPTION AND REFRACTION INDICES.
C
      DA=AC*ABS(AIX-AIO)
```

```
      X=3182.6018*ED/W/W
      Z=CF/W
      WRITE(2,1000) AZD,BN,ROR,RIO,F,HT
      WRITE(2,1100) ELD,BD,RXR,RIX,FRE,ED
      WRITE(2,1200) RLATD,BW,ROI,AIO,FR,X
      WRITE(2,1300) RLNGD,B,RXI,AIX,DAE,CF
      WRITE(2,1400) V,Y,CTH,TH,DA,Z
      WRITE(2,1500)
 1000 FORMAT("    AZ ",1PE13.6,"    BN ",1PE13.6,"    ROR ",1PE13.6,
     *       "   RIO ",1PE13.6,"     F ",1PE13.6,"     HT ",1PE13.6)
 1100 FORMAT("    EL ",1PE13.6,"    BD ",1PE13.6,"    RXR ",1PE13.6,
     *       "   RIX ",1PE13.6,"   FRE ",1PE13.6,"     ED ",1PE20.13)
 1200 FORMAT("   LAT ",1PE13.6,"    BW ",1PE13.6,"    ROI ",1PE13.6,
     *       "   AIO ",1PE13.6,"    FR ",1PE13.6,"      X ",1PE13.6)
 1300 FORMAT("   LNG ",1PE13.6,"     B ",1PE13.6,"    RXI ",1PE13.6,
     *       "   AIX ",1PE13.6,"   DAE ",1PE13.6,"     CF ",1PE13.6)
 1400 FORMAT("     V ",1PE13.6,"     Y ",1PE13.6,"    COS ",1PE13.6,
     *       "    TH ",1PE13.6,"    DA ",1PE13.6,"      Z ",1PE13.6)
 1500 FORMAT(" ")
      RETURN
      END
      SUBROUTINE COEFF(TM)
      DIMENSION G(11,11),GT(11,11),SHMIT(11,11)
      COMMON/COEFFS/TG(11,11)
C
C  READ SPHERICAL HARMONIC COEFFICIENTS.
C  THE COEFFICIENTS IN THE DATA STATEMENT ARE GSFC(12/66).
C
C  ARRAY 'G' CONTAINS BOTH G AND H VALUES.
C
       DATA G/
     *0.,-30401.2,-1540.1,1307.1,949.3,-233.5,49.2,72.2,8.5,10.4,-2.9,
     *5778.2,-2163.8,2997.9,-1988.9,803.5,355.7,57.5,-53.7,6.5,5.8,-.9,
     *-1932.,202.9,1590.3,1276.8,502.9,228.4,-.8,7.9,-9.3,7.5,-2.2,
     *-425.4,227.8,-133.8,881.2,-397.7,-28.8,-238.3,15.6,-9.6,-15.1,.8,
     *160.3,-274.3,2.3,-246.6,266.5,-157.9,-1.5,-24.3,-6.1,12.1,-2.8,
     *5.1,117.8,-114.8,-108.9,82.4,-62.2,-2.,-3.6,5.5,4.7,6.4,
     *-12.1,104.4,56.6,-23.4,-14.8,-13.3,-108.9,15.5,-8.1,.2,4.7,
     *-53.7,-27.4,-8.1,7.,24.3,-22.5,-21.4,3.6,13.,1.6,-.2,
     *5.4,-11.7,4.2,-15.3,4.6,21.9,-.7,-17.1,7.4,.9,1.8,
     *-22.4,13.8,6.3,-3.,-1.9,9.,11.5,.1,-1.5,.2,2.,
     *-.1,4.5,-1.,2.6,-4.4,-1.3,-3.6,4.,1.,-2.,1.1/
       DATA GT/
     *0.,14.03,-23.29,-.93,1.45,1.61,-.42,-.57,.35,-.10,-.01,
     *-3.71,8.76,-.09,-10.62,.9,.6,.82,-.34,.5,-.13,-.13,
     *-14.31,-16.62,-4.56,2.31,-1.75,3.34,.82,-1.44,1.7,-1.2,.88,
     *5.2,2.53,-6.98,-5.89,.66,-.04,2.35,-.9,-.11,.08,-.18,
     *-2.19,-.14,1.88,-6.52,-3.01,-.6,.83,.03,.34,-.08,.17,
     *2.24,1.59,-2.61,.5,-.12,1.76,.01,-.6,-.07,-.39,-.02,
     *.05,.09,2.55,-1.19,.33,.84,.23,-.17,.43,-.36,.05,
     *-.96,.01,.43,.75,-.33,.49,.9,-.64,-.15,.47,.17,
     *-.5,-.21,.03,-.79,.05,.1,-.36,-.43,-.42,.37,.16,
     *.66,.54,.03,.35,-.03,-.01,.45,-.05,.75,-.46,.31,
     *-.61,-.64,.02,.05,-.63,-.07,.07,-.03,-.02,-.45,-.23/
```

```
C
C  CALCULATE NORMALIZATION CONVERSION FACTORS.
C
      SHMIT(1,1)=-1.
      MAXN=11
      DO 10 N=2,MAXN
         SHMIT(N,1)=SHMIT(N-1,1)*FLOAT(2*N-3)/FLOAT(N-1)
         SHMIT(1,N)=0.
         JJ=2
         DO 10 M=2,N
            SHMIT(N,M)=SHMIT(N,M-1)*SQRT(FLOAT((N-M+1)*JJ)/FLOAT(N+M-2))
            SHMIT(M-1,N)=SHMIT(N,M)
            JJ=1
   10 CONTINUE
C
C  CONVERT COEFFICIENTS, SCHMIDT TO GAUSS.
C
      DO 20 N=2,MAXN
         DO 20 M=1,N
            G(N,M)=G(N,M)*SHMIT(N,M)
            GT(N,M)=GT(N,M)*SHMIT(N,M)
            IF(M.EQ.1) GOTO 20
            G(M-1,N)=G(M-1,N)*SHMIT(M-1,N)
            GT(M-1,N)=GT(M-1,N)*SHMIT(M-1,N)
   20 CONTINUE
C
C  CONVERT COEFFICIENTS TO NEW TIME.
C
      T=TM-1960.0
      DO 30 N=1,MAXN
         DO 30 M=1,N
            TG(N,M)=G(N,M)+T*GT(N,M)
            IF(M.EQ.1) GOTO 30
            TG(M-1,N)=G(M-1,N)+T*GT(M-1,N)
   30 CONTINUE
      RETURN
      END
      SUBROUTINE FIELD (R,ST,CT,SPH,CPH,BR,BT,BP,B)
C
C  THIS SUBROUTINE COMPUTES THE MAGNETIC FIELD AT THE ROCKET'S
C  LOCATION.
C
      DIMENSION P(11,11), DP(11,11), CONST(11,11)
      DIMENSION SP(11), CP(11), FN(11), FM(11)
      COMMON/COEFFS/G(11,11)
      NMAX=11
      P(1,1)=1.
      DP(1,1)=0.
      SP(1)=0.
      CP(1)=1.
      DO 10 N=2,11
         FN(N)=N
         DO 10 M=1,N
            FM(M)=M-1
```

```
            CONST(N,M)=FLOAT((N-2)**2-(M-1)**2)/FLOAT((2*N-3)*(2*N-5))
   10 CONTINUE
      SP(2)=SPH
      CP(2)=CPH
      DO 20 M=3,NMAX
         SP(M)=SP(2)*CP(M-1)+CP(2)*SP(M-1)
         CP(M)=CP(2)*CP(M-1)-SP(2)*SP(M-1)
   20 CONTINUE
      AOR=6.3712E6/R
      AR=AOR**2
      BT=0.
      BP=0.
      BR=0.
      DO 70 N=2,NMAX
         AR=AOR*AR
         DO 70 M=1,N
            IF(M.EQ.N) GOTO 30
            P(N,M)=CT*P(N-1,M)
            DP(N,M)=CT*DP(N-1,M)-ST*P(N-1,M)
            IF(M.EQ.N-1) GOTO 40
            P(N,M)=P(N,M)-CONST(N,M)*P(N-2,M)
            DP(N,M)=DP(N,M)-CONST(N,M)*DP(N-2,M)
            GOTO 40
   30       P(N,N)=ST*P(N-1,N-1)
            DP(N,N)=ST*DP(N-1,N-1)+CT*P(N-1,N-1)
   40       PAR=P(N,M)*AR
            IF(M.EQ.1) GOTO 50
            TEMP=G(N,M)*CP(M)+G(M-1,N)*SP(M)
            BP=BP-(G(N,M)*SP(M)-G(M-1,N)*CP(M))*FM(M)*PAR
            GOTO 60
   50       TEMP=G(N,M)*CP(M)
            BP=BP-(G(N,M)*SP(M))*FM(M)*PAR
   60       BT=BT+TEMP*DP(N,M)*AR
            BR=BR-TEMP*FN(N)*PAR
   70 CONTINUE
      BP=BP/ST
      B=SQRT(BT*BT+BP*BP+BR*BR)
      GTT=1.E-9
      BR=BR*GTT
      BT=BT*GTT
      BP=BP*GTT
      B=B*GTT
      RETURN
      END
      SUBROUTINE SENWYL(ED,CF,S,CPH,W,RIO,RIX,AIO,AIX,ROR,RXR,ROI,RXI)
C
C GENERALIZED MAGNETO-IONIC THEORY, SEN-WYLLER EQUATIONS.
C
      REAL ED,P2,W,CF,Q,R,T,S,A1,A2,A3,C32,C52,WMS,WPS,X
      COMPLEX EI,EII,EIII,AA,BB,CC,DD,EE,VC,VD,VE,VF,VG,VH,VO,VX
      COMPLEX CMPLX,CSQRT,BBVA,CCCPH,RO,RX
C
C DEFINE BURKE-HARA FUNCTIONS FOR C3/2 AND C5/2.
C
```

```
      C32(X)=(X*(X*(X*(X+24.653115)+113.9416)+11.287513)+.023983474)
     */(X*(X*(X*(X*(X*(X+24.656819)+120.49512)+289.58085)+149.21254)
     *+9.3877372)+.018064128)
      C52(X)=(X*(X*(X+6.6945939)+16.901002)+1.1630641)/(X*(X*(X*(X
     **(X+6.6314497)+35.355257)+68.920505)+64.093464)+4.3605732)
      P2=3182.6018*ED
      WPS=W+S
      WMS=W-S
      Q=P2/W/CF
      R=Q/CF
      T=2.5*Q
      A1=W/CF
      A2=WMS/CF
      A3=WPS/CF
      A=R*W*C32(A1)
      B=T*C52(A1)
      C=R*WMS*C32(A2)
      D=T*C52(A2)
      E=R*WPS*C32(A3)
      F=T*C52(A3)
      EI=CMPLX(1.-A,-B)
      EII=CMPLX(.5*(F-D),.5*(C-E))
      EIII=CMPLX(A-.5*(C+E),B-.5*(F+D))
      AA=2.*EI*(EI+EIII)
      BB=EIII*(EI+EIII)+EII**2
      CC=2.*EI*EII
      DD=2.*EI
      EE=2.*EIII
      VB=CPH*CPH
      VA=1.-VB
      VC=BB*BB*VA*VA-CC*CC*VB
      VD=CSQRT(VC)
      BBVA=BB*VA
      VE=AA+BBVA
      VF=DD+EE*VA
      VG=(VE+VD)/VF
      VH=(VE-VD)/VF
      VO=CSQRT(VG)
      VX=CSQRT(VH)
C
C  SEPARATE INDICES.
C
      RIO=REAL(VO)
      AIO=-AIMAG(VO)
      RIX=REAL(VX)
      AIX=-AIMAG(VX)
C
C  CALCULATE WAVE POLARIZATIONS.
C
      CCCPH=CC*CPH
      RO=-((BBVA-VD)/CCCPH)
      ROR=REAL(RO)
      ROI=AIMAG(RO)
      RX=-((BBVA+VD)/CCCPH)
```

```
      RXR=REAL(RX)
      RXI=AIMAG(RX)
      RETURN
      END


      SUBROUTINE PRESSR(LAT,ALT,DATE,PRES)
C
C  THIS SUBROUTINE FINDS THE PRESSURE (IN PASCALS) FOR THE
C  ALTITUDE RANGE 25KM TO 500KM.
C
C  THREE INPUT PARAMETERS ARE REQUIRED:
C
C     LAT --  THE LATITUDE OF THE POINT WHERE THE PRESSURE IS
C             TO BE COMPUTED. (ENTERED IN DEGREES)
C     ALT --  THE ALTITUDE OF THE POINT WHERE THE PRESSURE IS
C             TO BE COMPUTED. (ENTERED IN METERS)
C     DATE -  THE TIME OF YEAR.  THE DAY SHOULD BE ENTERED AS
C             A DECIMAL PART OF A YEAR, WITH THREE DIGIT
C             ACCURACY.  FOR EXAMPLE:  FEBRUARY 26 IS THE 57TH
C             DAY OF THE YEAR.  SO TO ENTER FEBRUARY 26, 1979,
C             DATE SHOULD BE EQUAL TO 1979.156.  (57/365=.156)
C
C  THE ONE OUTPUT VARIABLE IS THE PRESSURE, PRES.
C
C     IF THE ALTITUDE IS OUT OF BOUNDS, NEGATIVE VALUES ARE
C     RETURNED FOR THE PRESSURE. (IF ALT IS LESS THAN 25KM,
C     -1 IS RETURNED, IF ALT IS GREATER THAN 500KM, -2 IS
C     RETURNED.)
C
C  THIS SUBROUTINE USES THE DATA FROM THE CIRA 1972 (COSPAR
C  INTERNATIONAL REFERENCE ATMOSPHERE 1972) BOOK.  FROM 25KM
C  TO 110KM DATA IS USED FROM TABLE 24A.  FROM 120KM TO 500KM
C  DATA IS USED FROM TABLE  3.
C
C     ARRAY P(8,18,12) CONTAINS THE DATA FROM TABLE 24A,
C     ARRAY P1(81) CONTAINS THE DATA FROM TABLE 3, 120KM - 200KM,
C     ARRAY P2(151) CONTAINS THE DATA FROM TABLE 3, 200KM - 500KM.
C
C  A LOG INTERPOLATION IS DONE ON THESE DATA POINTS TO FIND
C  THE PRESSURE AT ANY ALTITUDE.
C
C  WRITTEN BY M K MCINERNEY AUGUST 1979.
C
      REAL LAT,DATE,ALT,PRES,P(8,18,12),P1(81),P2(151),NLAT,
     *NALT,LPRES,UPRES,LALT,UALT
      REAL PJAN(8,18),PFEB(8,18),PMAR(8,18),PAPR(8,18),PMAY(8,18),
     *PJUN(8,18),PJUL(8,18),PAUG(8,18),PSEP(8,18),POCT(8,18),
     *PNOV(8,18),PDEC(8,18),P2LOW(70),P2UP(81)
      INTEGER L,A,D
C
C  THE MAXIMUM NUMBER OF CONTINUATION CARDS WHICH CAN BE USED
C  IN FORTRAN IV IS LESS THAN WHAT WOULD BE REQUIRED TO ENTER
C  ARRAYS 'P' AND 'P2' IN THEIR ENTIRETY.  THEREFORE BREAK UP THESE
```

```
C  ARRAYS INTO SUBARRAYS.
C
      EQUIVALENCE (P(1,1,1),PJAN(1,1)),(P(1,1,2),PFEB(1,1)),
     *(P(1,1,3),PMAR(1,1)),(P(1,1,4),PAPR(1,1)),
     *(P(1,1,5),PMAY(1,1)),(P(1,1,6),PJUN(1,1)),
     *(P(1,1,7),PJUL(1,1)),(P(1,1,8),PAUG(1,1)),
     *(P(1,1,9),PSEP(1,1)),(P(1,1,10),POCT(1,1)),
     *(P(1,1,11),PNOV(1,1)),(P(1,1,12),PDEC(1,1)),
     *(P2(1),P2LOW(1)),(P2(71),P2UP(1))
C
C          ROWS ARE ALTITUDE IN 5 KM STEPS FROM 25 KM TO 110 KM.
C          COLUMNS ARE LATITUDE IN 10 DEGREE STEPS FROM 0 TO 70.
C
C  JANUARY 1
C
      DATA PJAN/
     *250.E01,247.E01,244.E01,239.E01,237.E01,241.E01,244.E01,240.E01,
     *118.E01,117.E01,114.E01,112.E01,111.E01,112.E01,111.E01,105.E01,
     *582.E00,576.E00,562.E00,546.E00,529.E00,527.E00,509.E00,472.E00,
     *299.E00,295.E00,286.E00,275.E00,261.E00,254.E00,237.E00,216.E00,
     *157.E00,155.E00,152.E00,146.E00,136.E00,127.E00,116.E00,103.E00,
     *842.E-1,832.E-1,816.E-1,778.E-1,720.E-1,659.E-1,587.E-1,515.E-1,
     *454.E-1,449.E-1,442.E-1,420.E-1,383.E-1,345.E-1,306.E-1,266.E-1,
     *241.E-1,237.E-1,232.E-1,217.E-1,198.E-1,177.E-1,155.E-1,134.E-1,
     *122.E-1,119.E-1,117.E-1,110.E-1,100.E-1,089.E-1,079.E-1,067.E-1,
     *577.E-2,561.E-2,549.E-2,519.E-2,481.E-2,435.E-2,382.E-2,317.E-2,
     *255.E-2,249.E-2,243.E-2,236.E-2,222.E-2,207.E-2,183.E-2,148.E-2,
     *110.E-2,108.E-2,105.E-2,102.E-2,100.E-2,095.E-2,084.E-2,067.E-2,
     *471.E-3,463.E-3,446.E-3,440.E-3,437.E-3,429.E-3,384.E-3,305.E-3,
     *197.E-3,194.E-3,187.E-3,184.E-3,190.E-3,191.E-3,174.E-3,138.E-3,
     *803.E-4,791.E-4,767.E-4,778.E-4,833.E-4,873.E-4,813.E-4,646.E-4,
     *350.E-4,345.E-4,338.E-4,345.E-4,379.E-4,401.E-4,376.E-4,301.E-4,
     *168.E-4,164.E-4,160.E-4,163.E-4,177.E-4,190.E-4,181.E-4,145.E-4,
     *898.E-5,889.E-5,856.E-5,843.E-5,899.E-5,939.E-5,887.E-5,701.E-5/
C
C  FEBRUARY 1
C
      DATA PFEB/
     *250.E01,246.E01,242.E01,239.E01,239.E01,243.E01,241.E01,231.E01,
     *118.E01,116.E01,114.E01,112.E01,111.E01,112.E01,111.E01,105.E01,
     *581.E00,572.E00,560.E00,543.E00,531.E00,523.E00,517.E00,485.E00,
     *298.E00,293.E00,286.E00,274.E00,264.E00,255.E00,246.E00,227.E00,
     *157.E00,156.E00,152.E00,144.E00,137.E00,130.E00,122.E00,110.E00,
     *848.E-1,839.E-1,820.E-1,773.E-1,734.E-1,683.E-1,625.E-1,546.E-1,
     *458.E-1,455.E-1,443.E-1,413.E-1,390.E-1,359.E-1,325.E-1,279.E-1,
     *243.E-1,238.E-1,231.E-1,215.E-1,202.E-1,185.E-1,165.E-1,140.E-1,
     *122.E-1,119.E-1,114.E-1,106.E-1,100.E-1,092.E-1,083.E-1,069.E-1,
     *577.E-2,551.E-2,534.E-2,504.E-2,482.E-2,450.E-2,397.E-2,321.E-2,
     *256.E-2,245.E-2,240.E-2,229.E-2,223.E-2,211.E-2,188.E-2,147.E-2,
     *111.E-2,106.E-2,105.E-2,102.E-2,101.E-2,097.E-2,086.E-2,066.E-2,
     *469.E-3,448.E-3,451.E-3,439.E-3,440.E-3,434.E-3,393.E-3,295.E-3,
     *200.E-3,188.E-3,188.E-3,187.E-3,191.E-3,192.E-3,175.E-3,133.E-3,
     *855.E-4,802.E-4,797.E-4,784.E-4,814.E-4,843.E-4,792.E-4,600.E-4,
     *395.E-4,364.E-4,362.E-4,357.E-4,366.E-4,377.E-4,359.E-4,278.E-4,
```

```
     *199.E-4,182.E-4,178.E-4,171.E-4,174.E-4,179.E-4,174.E-4,135.E-4,
     *113.E-4,101.E-4,096.E-4,091.E-4,091.E-4,093.E-4,090.E-4,071.E-4/
C
C  MARCH 1
C
      DATA PMAR/
     *251.E01,246.E01,244.E01,240.E01,240.E01,243.E01,238.E01,225.E01,
     *118.E01,117.E01,115.E01,113.E01,112.E01,112.E01,110.E01,105.E01,
     *582.E00,575.E00,566.E00,552.E00,533.E00,529.E00,521.E00,501.E00,
     *299.E00,297.E00,290.E00,279.E00,267.E00,262.E00,255.E00,242.E00,
     *159.E00,158.E00,154.E00,146.E00,139.E00,135.E00,130.E00,121.E00,
     *860.E-1,859.E-1,833.E-1,781.E-1,741.E-1,718.E-1,675.E-1,616.E-1,
     *465.E-1,461.E-1,446.E-1,416.E-1,392.E-1,380.E-1,354.E-1,319.E-1,
     *244.E-1,241.E-1,231.E-1,214.E-1,203.E-1,196.E-1,181.E-1,162.E-1,
     *121.E-1,117.E-1,113.E-1,106.E-1,101.E-1,098.E-1,090.E-1,079.E-1,
     *569.E-2,547.E-2,528.E-2,503.E-2,487.E-2,475.E-2,434.E-2,372.E-2,
     *255.E-2,240.E-2,237.E-2,231.E-2,225.E-2,223.E-2,205.E-2,171.E-2,
     *110.E-2,105.E-2,105.E-2,103.E-2,102.E-2,102.E-2,094.E-2,076.E-2,
     *476.E-3,444.E-3,450.E-3,446.E-3,438.E-3,444.E-3,420.E-3,339.E-3,
     *206.E-3,191.E-3,191.E-3,188.E-3,187.E-3,191.E-3,182.E-3,146.E-3,
     *903.E-4,811.E-4,802.E-4,789.E-4,785.E-4,804.E-4,783.E-4,645.E-4,
     *419.E-4,374.E-4,362.E-4,354.E-4,354.E-4,365.E-4,355.E-4,298.E-4,
     *213.E-4,183.E-4,175.E-4,169.E-4,167.E-4,174.E-4,177.E-4,153.E-4,
     *122.E-4,104.E-4,096.E-4,089.E-4,088.E-4,094.E-4,098.E-4,086.E-4/
C
C  APRIL 1
C
      DATA PAPR/
     *251.E01,250.E01,246.E01,244.E01,241.E01,241.E01,240.E01,239.E01,
     *119.E01,119.E01,117.E01,116.E01,114.E01,114.E01,113.E01,111.E01,
     *590.E00,588.E00,583.E00,570.E00,554.E00,553.E00,545.E00,522.E00,
     *305.E00,306.E00,300.E00,291.E00,282.E00,282.E00,273.E00,254.E00,
     *163.E00,162.E00,159.E00,153.E00,148.E00,149.E00,142.E00,130.E00,
     *886.E-1,883.E-1,861.E-1,825.E-1,801.E-1,804.E-1,761.E-1,686.E-1,
     *478.E-1,473.E-1,462.E-1,439.E-1,427.E-1,430.E-1,407.E-1,363.E-1,
     *250.E-1,247.E-1,240.E-1,229.E-1,223.E-1,225.E-1,210.E-1,188.E-1,
     *123.E-1,121.E-1,119.E-1,114.E-1,111.E-1,112.E-1,106.E-1,093.E-1,
     *575.E-2,568.E-2,565.E-2,551.E-2,539.E-2,546.E-2,511.E-2,448.E-2,
     *253.E-2,250.E-2,257.E-2,253.E-2,250.E-2,253.E-2,240.E-2,209.E-2,
     *110.E-2,109.E-2,113.E-2,114.E-2,112.E-2,114.E-2,108.E-2,093.E-2,
     *479.E-3,471.E-3,494.E-3,493.E-3,478.E-3,475.E-3,458.E-3,392.E-3,
     *212.E-3,206.E-3,213.E-3,212.E-3,198.E-3,192.E-3,184.E-3,158.E-3,
     *920.E-4,880.E-4,900.E-4,873.E-4,813.E-4,769.E-4,754.E-4,658.E-4,
     *416.E-4,386.E-4,386.E-4,380.E-4,359.E-4,348.E-4,341.E-4,305.E-4,
     *202.E-4,182.E-4,178.E-4,172.E-4,165.E-4,167.E-4,176.E-4,164.E-4,
     *112.E-4,098.E-4,093.E-4,089.E-4,086.E-4,092.E-4,102.E-4,099.E-4/
C
C  MAY 1
C
      DATA PMAY/
     *251.E01,254.E01,250.E01,249.E01,247.E01,246.E01,249.E01,254.E01,
     *120.E01,120.E01,120.E01,118.E01,117.E01,117.E01,118.E01,120.E01,
     *595.E00,598.E00,595.E00,584.E00,576.E00,577.E00,581.E00,586.E00,
     *309.E00,311.E00,307.E00,299.E00,298.E00,300.E00,299.E00,292.E00,
```

```
     *164.E00,165.E00,163.E00,159.E00,158.E00,160.E00,159.E00,155.E00,
     *889.E-1,896.E-1,882.E-1,859.E-1,861.E-1,880.E-1,869.E-1,832.E-1,
     *480.E-1,481.E-1,475.E-1,463.E-1,464.E-1,474.E-1,470.E-1,456.E-1,
     *253.E-1,254.E-1,248.E-1,242.E-1,244.E-1,250.E-1,248.E-1,239.E-1,
     *126.E-1,127.E-1,125.E-1,122.E-1,122.E-1,126.E-1,125.E-1,123.E-1,
     *582.E-2,597.E-2,600.E-2,587.E-2,591.E-2,607.E-2,608.E-2,596.E-2,
     *254.E-2,263.E-2,271.E-2,268.E-2,267.E-2,276.E-2,280.E-2,283.E-2,
     *111.E-2,115.E-2,118.E-2,115.E-2,115.E-2,118.E-2,121.E-2,121.E-2,
     *489.E-3,496.E-3,505.E-3,485.E-3,464.E-3,459.E-3,472.E-3,477.E-3,
     *213.E-3,214.E-3,215.E-3,199.E-3,181.E-3,169.E-3,170.E-3,170.E-3,
     *887.E-4,899.E-4,901.E-4,819.E-4,700.E-4,619.E-4,622.E-4,649.E-4,
     *379.E-4,379.E-4,382.E-4,351.E-4,308.E-4,276.E-4,280.E-4,292.E-4,
     *172.E-4,172.E-4,175.E-4,163.E-4,145.E-4,135.E-4,145.E-4,162.E-4,
     *091.E-4,088.E-4,089.E-4,084.E-4,078.E-4,079.E-4,089.E-4,100.E-4/
C
C  JUNE 1
C
      DATA PJUN/
     *251.E01,253.E01,253.E01,253.E01,256.E01,258.E01,260.E01,265.E01,
     *119.E01,120.E01,121.E01,121.E01,122.E01,123.E01,125.E01,128.E01,
     *592.E00,597.E00,597.E00,599.E00,602.E00,611.E00,627.E00,642.E00,
     *306.E00,308.E00,306.E00,308.E00,311.E00,318.E00,326.E00,330.E00,
     *161.E00,163.E00,162.E00,163.E00,166.E00,171.E00,176.E00,178.E00,
     *869.E-1,881.E-1,878.E-1,886.E-1,905.E-1,935.E-1,965.E-1,974.E-1,
     *468.E-1,474.E-1,474.E-1,477.E-1,487.E-1,508.E-1,528.E-1,537.E-1,
     *248.E-1,251.E-1,249.E-1,251.E-1,258.E-1,271.E-1,282.E-1,288.E-1,
     *125.E-1,127.E-1,126.E-1,125.E-1,128.E-1,137.E-1,144.E-1,150.E-1,
     *585.E-2,599.E-2,592.E-2,593.E-2,607.E-2,652.E-2,695.E-2,731.E-2,
     *255.E-2,265.E-2,264.E-2,262.E-2,265.E-2,286.E-2,313.E-2,335.E-2,
     *112.E-2,114.E-2,114.E-2,111.E-2,110.E-2,117.E-2,128.E-2,136.E-2,
     *486.E-3,487.E-3,479.E-3,456.E-3,423.E-3,423.E-3,453.E-3,483.E-3,
     *208.E-3,206.E-3,201.E-3,183.E-3,156.E-3,141.E-3,145.E-3,152.E-3,
     *853.E-4,852.E-4,833.E-4,737.E-4,577.E-4,480.E-4,489.E-4,526.E-4,
     *360.E-4,362.E-4,361.E-4,327.E-4,260.E-4,214.E-4,216.E-4,230.E-4,
     *162.E-4,164.E-4,167.E-4,156.E-4,127.E-4,108.E-4,112.E-4,124.E-4,
     *840.E-5,837.E-5,867.E-5,836.E-5,719.E-5,640.E-5,689.E-5,771.E-5/
C
C  JULY 1
C
      DATA PJUL/
     *250.E01,252.E01,256.E01,259.E01,263.E01,265.E01,269.E01,274.E01,
     *118.E01,119.E01,122.E01,123.E01,125.E01,128.E01,130.E01,133.E01,
     *582.E00,591.E00,600.E00,605.E00,622.E00,640.E00,653.E00,662.E00,
     *299.E00,301.E00,305.E00,309.E00,318.E00,331.E00,340.E00,345.E00,
     *157.E00,158.E00,161.E00,163.E00,169.E00,177.E00,184.E00,185.E00,
     *084.E00,085.E00,087.E00,088.E00,091.E00,096.E00,101.E00,102.E00,
     *454.E-1,457.E-1,466.E-1,470.E-1,492.E-1,521.E-1,551.E-1,564.E-1,
     *241.E-1,242.E-1,245.E-1,246.E-1,257.E-1,275.E-1,295.E-1,307.E-1,
     *122.E-1,123.E-1,123.E-1,122.E-1,128.E-1,139.E-1,152.E-1,159.E-1,
     *577.E-2,578.E-2,573.E-2,562.E-2,595.E-2,654.E-2,730.E-2,779.E-2,
     *255.E-2,256.E-2,253.E-2,245.E-2,257.E-2,284.E-2,322.E-2,348.E-2,
     *110.E-2,110.E-2,109.E-2,104.E-2,104.E-2,111.E-2,126.E-2,137.E-2,
     *471.E-3,471.E-3,460.E-3,423.E-3,395.E-3,393.E-3,425.E-3,452.E-3,
     *197.E-3,198.E-3,191.E-3,170.E-3,145.E-3,131.E-3,131.E-3,135.E-3,
```

```
     *803.E-4,812.E-4,793.E-4,688.E-4,569.E-4,479.E-4,458.E-4,456.E-4,
     *350.E-4,357.E-4,354.E-4,317.E-4,262.E-4,218.E-4,198.E-4,191.E-4,
     *168.E-4,170.E-4,168.E-4,153.E-4,132.E-4,110.E-4,101.E-4,098.E-4,
     *898.E-5,903.E-5,895.E-5,835.E-5,725.E-5,626.E-5,576.E-5,559.E-5/
C
C  AUGUST 1
C
      DATA PAUG/
     *250.E01,253.E01,257.E01,260.E01,264.E01,265.E01,270.E01,276.E01,
     *118.E01,120.E01,122.E01,123.E01,126.E01,128.E01,130.E01,132.E01,
     *581.E00,592.E00,595.E00,606.E00,623.E00,640.E00,648.E00,650.E00,
     *298.E00,301.E00,303.E00,307.E00,319.E00,330.E00,335.E00,332.E00,
     *157.E00,158.E00,158.E00,161.E00,169.E00,175.E00,178.E00,177.E00,
     *848.E-1,844.E-1,852.E-1,863.E-1,907.E-1,946.E-1,971.E-1,963.E-1,
     *458.E-1,457.E-1,454.E-1,462.E-1,485.E-1,509.E-1,526.E-1,525.E-1,
     *243.E-1,240.E-1,240.E-1,241.E-1,252.E-1,266.E-1,280.E-1,281.E-1,
     *122.E-1,123.E-1,121.E-1,120.E-1,124.E-1,132.E-1,141.E-1,145.E-1,
     *577.E-2,584.E-2,579.E-2,563.E-2,571.E-2,615.E-2,676.E-2,709.E-2,
     *256.E-2,268.E-2,261.E-2,251.E-2,250.E-2,268.E-2,295.E-2,316.E-2,
     *111.E-2,116.E-2,116.E-2,109.E-2,104.E-2,106.E-2,115.E-2,125.E-2,
     *469.E-3,506.E-3,501.E-3,459.E-3,408.E-3,390.E-3,397.E-3,430.E-3,
     *200.E-3,214.E-3,214.E-3,189.E-3,158.E-3,140.E-3,134.E-3,138.E-3,
     *855.E-4,931.E-4,902.E-4,796.E-4,660.E-4,567.E-4,504.E-4,489.E-4,
     *395.E-4,421.E-4,412.E-4,359.E-4,301.E-4,255.E-4,216.E-4,199.E-4,
     *199.E-4,213.E-4,199.E-4,171.E-4,145.E-4,122.E-4,102.E-4,091.E-4,
     *113.E-4,117.E-4,109.E-4,090.E-4,075.E-4,063.E-4,052.E-4,045.E-4/
C
C  SEPTEMBER 1
C
      DATA PSEP/
     *251.E01,253.E01,254.E01,257.E01,259.E01,260.E01,260.E01,262.E01,
     *118.E01,120.E01,121.E01,122.E01,124.E01,124.E01,124.E01,124.E01,
     *582.E00,589.E00,596.E00,597.E00,607.E00,613.E00,610.E00,599.E00,
     *299.E00,301.E00,303.E00,302.E00,307.E00,313.E00,310.E00,301.E00,
     *159.E00,159.E00,160.E00,158.E00,162.E00,165.E00,163.E00,157.E00,
     *860.E-1,858.E-1,858.E-1,845.E-1,865.E-1,883.E-1,875.E-1,840.E-1,
     *465.E-1,464.E-1,461.E-1,451.E-1,461.E-1,471.E-1,467.E-1,450.E-1,
     *244.E-1,245.E-1,242.E-1,234.E-1,235.E-1,241.E-1,243.E-1,237.E-1,
     *121.E-1,124.E-1,125.E-1,118.E-1,115.E-1,117.E-1,121.E-1,120.E-1,
     *569.E-2,600.E-2,606.E-2,560.E-2,528.E-2,537.E-2,573.E-2,582.E-2,
     *255.E-2,275.E-2,283.E-2,255.E-2,233.E-2,234.E-2,252.E-2,262.E-2,
     *110.E-2,122.E-2,126.E-2,112.E-2,098.E-2,096.E-2,103.E-2,110.E-2,
     *476.E-3,527.E-3,550.E-3,477.E-3,405.E-3,378.E-3,399.E-3,430.E-3,
     *206.E-3,230.E-3,236.E-3,200.E-3,166.E-3,150.E-3,153.E-3,164.E-3,
     *090.E-3,101.E-3,104.E-3,087.E-3,072.E-3,063.E-3,062.E-3,064.E-3,
     *419.E-4,480.E-4,487.E-4,404.E-4,328.E-4,280.E-4,265.E-4,261.E-4,
     *213.E-4,246.E-4,247.E-4,199.E-4,158.E-4,131.E-4,119.E-4,110.E-4,
     *122.E-4,141.E-4,138.E-4,107.E-4,082.E-4,066.E-4,056.E-4,048.E-4/
C
C  OCTOBER 1
C
      DATA POCT/
     *251.E01,250.E01,251.E01,253.E01,254.E01,254.E01,250.E01,243.E01,
     *119.E01,119.E01,119.E01,120.E01,119.E01,118.E01,116.E01,113.E01,
```

```
     *590.E00,588.E00,585.E00,583.E00,578.E00,564.E00,556.E00,539.E00,
     *305.E00,302.E00,298.E00,295.E00,289.E00,281.E00,274.E00,264.E00,
     *163.E00,161.E00,157.E00,153.E00,150.E00,145.E00,140.E00,135.E00,
     *886.E-1,869.E-1,850.E-1,824.E-1,801.E-1,767.E-1,740.E-1,708.E-1,
     *478.E-1,468.E-1,457.E-1,439.E-1,426.E-1,403.E-1,389.E-1,374.E-1,
     *250.E-1,246.E-1,240.E-1,228.E-1,217.E-1,204.E-1,199.E-1,191.E-1,
     *123.E-1,123.E-1,122.E-1,113.E-1,106.E-1,099.E-1,099.E-1,095.E-1,
     *575.E-2,586.E-2,594.E-2,549.E-2,487.E-2,454.E-2,466.E-2,452.E-2,
     *253.E-2,265.E-2,274.E-2,248.E-2,217.E-2,202.E-2,213.E-2,207.E-2,
     *110.E-2,117.E-2,122.E-2,110.E-2,094.E-2,087.E-2,093.E-2,092.E-2,
     *479.E-3,515.E-3,533.E-3,471.E-3,402.E-3,371.E-3,398.E-3,400.E-3,
     *212.E-3,228.E-3,233.E-3,202.E-3,172.E-3,158.E-3,169.E-3,170.E-3,
     *092.E-3,099.E-3,102.E-3,088.E-3,075.E-3,069.E-3,074.E-3,074.E-3,
     *416.E-4,462.E-4,480.E-4,413.E-4,343.E-4,310.E-4,325.E-4,316.E-4,
     *202.E-4,230.E-4,242.E-4,207.E-4,172.E-4,151.E-4,150.E-4,139.E-4,
     *112.E-4,130.E-4,137.E-4,116.E-4,093.E-4,078.E-4,072.E-4,061.E-4/
C
C  NOVEMBER 1
C
      DATA PNOV/
     *251.E01,250.E01,248.E01,247.E01,246.E01,245.E01,244.E01,239.E01,
     *120.E01,119.E01,118.E01,116.E01,115.E01,113.E01,111.E01,109.E01,
     *595.E00,589.E00,578.E00,562.E00,546.E00,529.E00,518.E00,511.E00,
     *309.E00,303.E00,294.E00,283.E00,270.E00,255.E00,247.E00,244.E00,
     *164.E00,161.E00,155.E00,148.E00,139.E00,129.E00,123.E00,122.E00,
     *889.E-1,869.E-1,838.E-1,792.E-1,736.E-1,666.E-1,633.E-1,632.E-1,
     *480.E-1,470.E-1,452.E-1,425.E-1,388.E-1,346.E-1,330.E-1,333.E-1,
     *253.E-1,248.E-1,239.E-1,221.E-1,200.E-1,177.E-1,168.E-1,169.E-1,
     *126.E-1,124.E-1,120.E-1,110.E-1,098.E-1,087.E-1,084.E-1,084.E-1,
     *582.E-2,576.E-2,571.E-2,525.E-2,464.E-2,413.E-2,404.E-2,396.E-2,
     *254.E-2,255.E-2,256.E-2,237.E-2,210.E-2,191.E-2,190.E-2,184.E-2,
     *111.E-2,112.E-2,113.E-2,104.E-2,093.E-2,086.E-2,087.E-2,084.E-2,
     *489.E-3,495.E-3,492.E-3,448.E-3,402.E-3,381.E-3,393.E-3,387.E-3,
     *213.E-3,216.E-3,212.E-3,192.E-3,175.E-3,169.E-3,178.E-3,176.E-3,
     *887.E-4,899.E-4,888.E-4,826.E-4,774.E-4,772.E-4,819.E-4,811.E-4,
     *379.E-4,395.E-4,405.E-4,384.E-4,365.E-4,364.E-4,386.E-4,372.E-4,
     *172.E-4,184.E-4,195.E-4,192.E-4,183.E-4,180.E-4,184.E-4,174.E-4,
     *091.E-4,100.E-4,109.E-4,106.E-4,099.E-4,093.E-4,091.E-4,081.E-4/
C
C  DECEMBER 1
C
      DATA PDEC/
     *251.E01,247.E01,245.E01,241.E01,239.E01,241.E01,242.E01,240.E01,
     *119.E01,118.E01,116.E01,113.E01,111.E01,112.E01,110.E01,107.E01,
     *592.E00,584.E00,565.E00,552.E00,530.E00,524.E00,506.E00,486.E00,
     *306.E00,301.E00,289.E00,279.E00,263.E00,252.E00,237.E00,227.E00,
     *161.E00,159.E00,152.E00,147.E00,136.E00,125.E00,116.E00,110.E00,
     *869.E-1,854.E-1,827.E-1,788.E-1,723.E-1,645.E-1,586.E-1,563.E-1,
     *468.E-1,460.E-1,442.E-1,422.E-1,382.E-1,336.E-1,303.E-1,294.E-1,
     *248.E-1,243.E-1,234.E-1,221.E-1,198.E-1,172.E-1,154.E-1,150.E-1,
     *125.E-1,122.E-1,117.E-1,111.E-1,099.E-1,086.E-1,078.E-1,075.E-1,
     *585.E-2,568.E-2,556.E-2,533.E-2,477.E-2,420.E-2,381.E-2,357.E-2,
     *255.E-2,250.E-2,245.E-2,239.E-2,220.E-2,199.E-2,183.E-2,167.E-2,
     *112.E-2,109.E-2,108.E-2,105.E-2,099.E-2,091.E-2,085.E-2,077.E-2,
```

```
     *486.E-3,477.E-3,459.E-3,449.E-3,433.E-3,411.E-3,388.E-3,353.E-3,
     *208.E-3,204.E-3,199.E-3,194.E-3,191.E-3,185.E-3,179.E-3,162.E-3,
     *853.E-4,833.E-4,807.E-4,822.E-4,849.E-4,867.E-4,841.E-4,764.E-4,
     *360.E-4,353.E-4,353.E-4,368.E-4,393.E-4,412.E-4,405.E-4,361.E-4,
     *162.E-4,161.E-4,161.E-4,173.E-4,188.E-4,201.E-4,195.E-4,172.E-4,
     *840.E-5,850.E-5,872.E-5,912.E-5,962.E-5,992.E-5,947.E-5,816.E-5/
C
C  PRESSURES FROM 120 KM TO 200 KM IN 1 KM STEPS.
C
      DATA P1/
     *2.667E-3,2.449E-3,2.257E-3,2.085E-3,1.932E-3,1.795E-3,
     *1.671E-3,1.560E-3,1.459E-3,1.367E-3,1.283E-3,1.207E-3,1.137E-3,
     *1.073E-3,1.014E-3,9.597E-4,9.094E-4,8.629E-4,8.198E-4,7.797E-4,
     *7.423E-4,7.075E-4,6.749E-4,6.445E-4,6.159E-4,5.891E-4,5.640E-4,
     *5.403E-4,5.179E-4,4.969E-4,4.770E-4,4.582E-4,4.404E-4,4.235E-4,
     *4.075E-4,3.923E-4,3.778E-4,3.641E-4,3.510E-4,3.386E-4,3.267E-4,
     *3.154E-4,3.046E-4,2.943E-4,2.844E-4,2.749E-4,2.659E-4,2.573E-4,
     *2.490E-4,2.410E-4,2.334E-4,2.261E-4,2.191E-4,2.123E-4,2.058E-4,
     *1.996E-4,1.936E-4,1.878E-4,1.823E-4,1.769E-4,1.718E-4,1.668E-4,
     *1.620E-4,1.574E-4,1.529E-4,1.486E-4,1.445E-4,1.405E-4,1.366E-4,
     *1.329E-4,1.292E-4,1.257E-4,1.224E-4,1.191E-4,1.159E-4,1.129E-4,
     *1.099E-4,1.070E-4,1.043E-4,1.016E-4,9.896E-5/
C
C  PRESSURES FROM 200 KM TO 500 KM IN 2 KM STEPS.
C
      DATA P2LOW/
     *9.896E-5,9.399E-5,8.931E-5,8.491E-5,8.076E-5,7.686E-5,7.317E-5,
     *6.970E-5,6.642E-5,6.332E-5,6.039E-5,5.762E-5,5.499E-5,5.251E-5,
     *5.015E-5,4.792E-5,4.580E-5,4.379E-5,4.188E-5,4.006E-5,3.834E-5,
     *3.670E-5,3.514E-5,3.366E-5,3.224E-5,3.090E-5,2.961E-5,2.839E-5,
     *2.723E-5,2.611E-5,2.505E-5,2.404E-5,2.307E-5,2.215E-5,2.127E-5,
     *2.042E-5,1.962E-5,1.885E-5,1.811E-5,1.740E-5,1.673E-5,1.608E-5,
     *1.546E-5,1.487E-5,1.430E-5,1.376E-5,1.324E-5,1.274E-5,1.226E-5,
     *1.180E-5,1.136E-5,1.093E-5,1.053E-5,1.014E-5,9.764E-6,9.405E-6,
     *9.061E-6,8.730E-6,8.412E-6,8.106E-6,7.812E-6,7.530E-6,7.259E-6,
     *6.998E-6,6.747E-6,6.507E-6,6.275E-6,6.052E-6,5.837E-6,5.631E-6/
      DATA P2UP/
     *5.432E-6,5.241E-6,5.057E-6,4.880E-6,4.709E-6,4.545E-6,4.387E-6,
     *4.235E-6,4.088E-6,3.947E-6,3.811E-6,3.680E-6,3.554E-6,3.432E-6,
     *3.315E-6,3.202E-6,3.093E-6,2.988E-6,2.886E-6,2.789E-6,2.694E-6,
     *2.604E-6,2.516E-6,2.432E-6,2.350E-6,2.272E-6,2.196E-6,2.123E-6,
     *2.052E-6,1.984E-6,1.918E-6,1.855E-6,1.794E-6,1.735E-6,1.677E-6,
     *1.622E-6,1.569E-6,1.518E-6,1.468E-6,1.421E-6,1.374E-6,1.330E-6,
     *1.287E-6,1.245E-6,1.205E-6,1.166E-6,1.129E-6,1.092E-6,1.057E-6,
     *1.024E-6,9.910E-7,9.595E-7,9.290E-7,8.995E-7,8.710E-7,8.435E-7,
     *8.169E-7,7.912E-7,7.663E-7,7.423E-7,7.191E-7,6.966E-7,6.749E-7,
     *6.539E-7,6.337E-7,6.140E-7,5.950E-7,5.767E-7,5.590E-7,5.418E-7,
     *5.252E-7,5.092E-7,4.936E-7,4.786E-7,4.641E-7,4.500E-7,4.364E-7,
     *4.233E-7,4.106E-7,3.982E-7,3.863E-7/
C
C  INITIALIZE ARRAY INDICES.
C
      L=0
      A=0
```

```
      D=0
C
C  BRANCH TO APPROPRIATE ALTITUDE SECTION.
C
      IF(ALT.LT.25000.) GOTO 50
      IF(ALT.GE.120000.) GOTO 20
C
C    ALTITUDE RANGE 25-110 KM.
C
C  FIND WHICH POSITION IN THE ARRAY TO USE FOR THE LATITUDE.
C
      NLAT=ABS(LAT/10.)
      CALL ROUND(NLAT,L)
      L=L+1
      IF(L.GT.8) L=8
C
C  FIND WHICH POSITION IN THE ARRAY TO USE FOR ALTITUDE.
C
      A=INT((ALT-20000.)/5000.)
      IF(A.GT.18) A=18
C
C  FIND WHICH POSITION IN THE ARRAY TO USE FOR THE MONTH.
C
      CALL MONTH(DATE,D)
C
C  SET THE VALUES OF THE PRESSURE AND ALTITUDE WHICH BRACKET THE
C  ALTITUDE AT WHICH THE PRESSURE IS TO BE COMPUTED.
C
      LPRES=P(L,A,D)
      LALT=(FLOAT(A)*5000.)+20000.
      IF(A.EQ.18) GO TO 10
      UPRES=P(L,(A+1),D)
      UALT=(FLOAT(A+1)*5000.)+20000.
      CALL INTER(LPRES,UPRES,LALT,UALT,ALT,PRES)
      RETURN
C
C  FOR ALTITUDES BETWEEN 110 KM AND 120 KM,
C  THE UPPER PRESSURE VALUE WILL BE FOUND IN ARRAY 'P1'.
C
   10 UPRES=P1(1)
      UALT=120000.
      CALL INTER(LPRES,UPRES,LALT,UALT,ALT,PRES)
      RETURN
C
C  FOR THE ALTITUDE RANGE 120-200 KM, GO THROUGH ALMOST THE
C  SAME PROCESS AS ABOVE, ONLY WITH NO VARIATION IN LATITUDE
C  OR TIME OF YEAR.
C
   20 IF(ALT.GE.200000.) GOTO 30
      NALT=((ALT-119000.)/1000.)
      A=INT(NALT)
      LPRES=P1(A)
      LALT=(FLOAT(A)*1000.)+119000.
      UPRES=P1(A+1)
```

```
      UALT=(FLOAT(A+1)*1000.)+119000.
      CALL INTER(LPRES,UPRES,LALT,UALT,ALT,PRES)
      RETURN
C
C    ALTITUDE RANGE 200-500 KM.
C
   30 IF(ALT.GE.500000.) GOTO 40
      NALT=((ALT-198000.)/2000.)
      A=INT(NALT)
      LPRES=P2(A)
      LALT=(FLOAT(A)*2000.)+198000.
      UPRES=P2(A+1)
      UALT=(FLOAT(A+1)*2000.)+198000.
      CALL INTER(LPRES,UPRES,LALT,UALT,ALT,PRES)
      RETURN
   40 IF(ALT.GT.500000.) GO TO 60
      PRES=P2(151)
      RETURN
C
C  SET PRESSURE EQUAL TO -1 IF ALTITUDE IS LESS THAN 25 KM.
C
   50 PRES=-1.0
      RETURN
C
C  SET PRESSURE EQUAL TO -2 IF ALTITUDE IS GREATER THAN 500 KM.
C
   60 PRES=-2.0
      RETURN
      END
      SUBROUTINE MONTH(DATE,D)
C
C  THIS SUBROUTINE FINDS THE MONTH WHICH STARTS CLOSEST TO THE
C  DATE ENTERED.
C
      REAL YEAR,NDATE,DATE
      INTEGER DAY,Z,M(13),D
      DATA M/1,32,60,91,121,152,182,213,244,274,305,335,366/
      Z=0
C
C  SEPARATE YEAR.
C
      YEAR=AINT(DATE)
C
C  COMPUTE CORRECTION FOR LEAP YEAR.
C
      IF(MOD(INT(YEAR),2).EQ.1) GOTO 10
      NDATE=YEAR/2.
      Z=MOD(INT(NDATE),2)
C
C  COMPUTE DAY OF YEAR.
C
   10 NDATE=(DATE-YEAR)*(365.+FLOAT(Z))
      CALL ROUND(NDATE,DAY)
      IF(DAY.LT.(M(3)+Z)) GOTO 50
```

```
      NDATE=FLOAT(DAY)
C
C  ESTIMATE MONTH IN WHICH DAY OCCURS.
C
      D=INT(NDATE/30.416667)+1
C
C  ITERATE TO FIND THE EXACT MONTH IN WHICH THE DAY OCCURS.
C
   20 IF(DAY.GE.(M(D)+Z)) GO TO 30
      D=D-1
      GO TO 20
   30 IF(DAY.LT.(M(D+1)+Z)) GO TO 40
      D=D+1
      GO TO 30
C
C  FIND FIRST OF MONTH WHICH IS CLOSEST TO DAY.
C
   40 IF((M(D+1)+Z-DAY).LT.(DAY-M(D)-Z)) D=D+1
      IF(D.EQ.13) D=1
      RETURN
C
C  FIND FIRST OF MONTH WHICH IS CLOSEST TO DAY WHEN
C  DAY IS IN FEBRUARY.
C
   50 IF(DAY.LE.M(2)) GO TO 60
      D=2
      IF((M(3)+Z-DAY).LT.(DAY-M(2))) D=3
      RETURN
C
C  FIND FIRST OF MONTH WHICH IS CLOSEST TO DAY WHEN
C  DAY IS IN JANUARY.
C
   60 D=1
      IF((M(2)-DAY).LT.(DAY-M(1))) D=2
      RETURN
      END
      SUBROUTINE INTER(LPRES,UPRES,LALT,UALT,ALT,PRES)
C
C  THIS SUBROUTINE DOES A LOG INTERPOLATION BETWEEN TWO PRESSURE
C  VALUES.
C
      REAL LPRES,UPRES,LALT,UALT,ALT,PRES
      IF(LPRES.EQ.0..OR.UPRES.EQ.0.) GO TO 10
      PRES=EXP((((ALT-LALT)/(UALT-LALT))*(ALOG(UPRES)-ALOG(LPRES)))+ALOG
     *(LPRES))
      RETURN
   10 PRES=0.0
      RETURN
      END
      SUBROUTINE ROUND(RI,IO)
C
C  THIS SUBROUTINE ROUNDS OFF A DECIMAL NUMBER TO THE NEAREST
C  INTEGER.
C
```

```
      REAL RI,A
      INTEGER IO,B
      IO=INT(RI)
      A=RI-FLOAT(IO)
      IF(A.GT..5) IO=IO+1
      IF(A.LT.-.5) IO=IO-1
      IF(A.EQ..5) GO TO 10
      IF(A.EQ.-.5) GO TO 20
      RETURN
   10 IF(MOD(IO,2).EQ.0) IO=IO+1
      RETURN
   20 IF(MOD(IO,2).EQ.0) IO=IO-1
      RETURN
      END
```

APPENDIX VI. Listing of IBM Fortran IV program DACAL.

The JCL necessary to compile and execute this program on the IBM is included. One minute and fifteen seconds (1650:05UT → 1651:20UT) from the 18.1020 differential absorption data tape, named WI20DA, was processed using this listing.

Note that within program DACAL time is stored as an integer, in units of tenths of milliseconds. This is done to avoid round-off errors. Since IBM single precision floating-point numbers are only 32 bits long, giving an accuracy of approximately 7.2 decimal digits, round-off errors will propagate inaccuracies in the final decimal places. This is unacceptable because the final decimal places are important to the process of locating record indices. (The maximum time possible is 2359:59.9999 seconds which, in floating-point notation, is .863999999E+05.)

```
//MIKE1 JOB
/*ID PS=2714,NAME='MCINERNEY'
/*ID CODE=
/*ID PRINT=LOCAL
/*ID BIN=49
/*ID EJECT=YES
/*ID IOREQ=2000,LINES=5000,TIME=(2,00)
/*SETUP UNIT=TAPE,R=C383,ID=(WI20DA,NORING)
// EXEC FORTLDGO
//FORT.SYSIN DD *
C                         ----- PROGRAM DACAL -----
C
C  THIS PROGRAM IS USED TO COMPUTE THE DIFFERENTIAL ABSORPTION
C  CALIBRATION LEVELS FROM DATA TAPES DIGITIZED AT THE WALLOPS
C  PCM FACILITY.
C
C  THE FIRST FOUR RECORDS ON THE TAPE ARE HEADER RECORDS.
C  THE NEXT FIVE RECORDS ARE DIGITIZER CALIBRATION RECORDS.
C  THE REMAINING RECORDS ARE DATA RECORDS.
C
C  THE PROGRAM IS DESIGNED TO PROCESS DATA RECORDS 2008 WORDS
C  LONG.  THE FIRST FIVE WORDS ARE A HEADER.  THE LAST THREE
C  WORDS CONTAIN TIME INFORMATION.  EACH DATA FRAME CONSISTS
C  OF FIVE CHANNELS.
C
C  THE DIGITIZATION RATE IS ASSUMED TO BE 5000 SAMPLES/SECOND
C  RESULTING IN A SAMPLING RATE OF 1000 SAMPLES/CHANNEL/SECOND.
C
C  THERE ARE TWO INPUT CARDS:  THE TIME TO BEGIN PROCESSING AND
C  THE TIME TO END PROCESSING.  THEY ARE CODED IN THE FOLLOWING
C  FORMAT:
C
C         HHMM:SECS       (2I2,1X,F5.2)
C
C      COLUMN 1
C
```

```
C  THIS PROGRAM AVERAGES EVERY FIFTY DATA POINTS AND PRINTS OUT
C  THIS AVERAGE FOR EACH OF THE FIVE CHANNELS.
C  RECORD TIMES ARE PRINTED OUT ONCE A SECOND FOR DIAGNOSTIC
C  PURPOSES (E.G. TO DETERMINE IF THE TIME CODE IS "SLIPPING")
C  AND LOCATION DETERMINATION.
C
C  MODIFIED BY M.K. MCINERNEY OCTOBER, 1979.
C
      INTEGER*2 I(2100),K(10)
      INTEGER ENDTIM,EOF,ERRAD,ERRCD,ISUM(5),RCDHRS,RCDMIN,RCDSEC,
     *RCDTIM,STARTM,TIMDIF,TIMSEC
      REAL AVRIJ(5),SCALE
      COMMON RCDTIM,NXTTIM,RCDHRS,NXTHRS,RCDMIN,NXTMIN,RCDSEC,NXTSEC,I
      ASSIGN 150 TO ERRAD
      ASSIGN 180 TO EOF
C
C  READ START AND STOP TIMES, CONVERT TO TENTHS OF MILLISECONDS
C  AND PRINT OUT FOR DIAGNOSTIC PURPOSES.
C
      READ(5,1000) ISTHRS,ISTMNS,STSECS
 1000 FORMAT(2I2,1X,F5.2)
      STARTM=TIMSEC(ISTHRS,ISTMNS,STSECS)
      READ(5,1000) IENDHR,IENDMN,ENDSEC
      ENDTIM=TIMSEC(IENDHR,IENDMN,ENDSEC)
      WRITE(6,1100) ISTHRS,ISTMNS,STSECS,IENDHR,IENDMN,ENDSEC
 1100 FORMAT('1','START TIME = ',2I2,':',F5.2,5X,'END TIME = ',
     *2I2,':',F5.2/'0')
C
C  OPEN TAPE FILE AND INITIALIZE VARIABLES.
C
      CALL TPOPIZ(12,ERRAD,ERRCD)
      IFLAG=0
      NBLOCK=0
      NLINES=0
      NXTHRS=0
      NXTMIN=0
      NXTSEC=0
      NXTTIM=0
C
C  SKIP OVER THE FOUR HEADER AND FIVE CALIBRATION RECORDS
C  AT THE BEGINNING OF THE TAPE.
C
      DO 10 J=1,9
         CALL TPFSRZ(12)
         NBLOCK=NBLOCK+1
   10 CONTINUE
      IFLAG=1
C
C  FETCH FIRST DATA RECORD, COMPUTE START AND END TIMES
C  OF RECORD AND PRINT OUT TIMES FOR DIAGNOSTIC PURPOSES.
C
   20 CALL TPGETZ(12,I)
      CALL TPCHKZ(12,NBYTES,EOF)
      NBLOCK=NBLOCK+1
```

```
      CALL GETIME
      RCDSEC=NXTSEC-4000
      RCDTIM=NXTTIM-4000
      TEMP1=SCALE(RCDSEC)
      WRITE(6,1200) NBLOCK,NXTHRS,NXTMIN,TEMP1
 1200 FORMAT(' RECORD ',I5,' STARTS AT TIME ',2I2,':',F7.4)
      IFLAG=2
      IF(RCDTIM.GT.ENDTIM) STOP
      IF(NXTTIM.GE.STARTM) GOTO 40
C
C  ADVANCE ON TAPE UNTIL THE RECORD CONTAINING THE STARTING
C  TIME IS REACHED.
C
   30 CALL TPGETZ(12,I)
      CALL TPCHKZ(12,NBYTES,EOF)
      NBLOCK=NBLOCK+1
      CALL GETIME
      TEMP1=SCALE(RCDSEC)
      WRITE(6,1200) NBLOCK,RCDHRS,RCDMIN,TEMP1
      IF(RCDTIM.GT.ENDTIM) STOP
      IF(NXTTIM.LT.STARTM) GOTO 30
C
C  CALCULATE THE FRAME WHICH CONTAINS THE STARTING TIME.
C
   40 IFLAG=3
      TIMDIF=STARTM-RCDTIM
      INDEX=INT(FLOAT(TIMDIF*2000)/FLOAT(NXTTIM-RCDTIM))+5
      INDEX=INDEX-MOD(INDEX,5)
      TEMP1=SCALE(RCDTIM)
      TEMP2=SCALE(NXTTIM)
      TEMP3=SCALE(STARTM)
      TEMP4=SCALE(TIMDIF)
      WRITE(6,1300) TEMP1,TEMP2,TEMP3,TEMP4,INDEX
 1300 FORMAT(' '/'0   RECORD STARTS AT ',F11.4,'  RECORD ENDS AT ',
     *F11.4,'  PROCESSING IS TO START AT ',F11.4/'ORECORD DISPLACEMENT '
     *,F11.4,'  STARTING INDEX ',I4/)
C
C  SUM CHANNELS.
C  COMPUTE AVERAGE EVERY FIFTY SAMPLES AND PRINT OUT.
C  PRINT OUT RECORD NUMBER AND TIME EVERY SECOND
C  (1000 DATA SAMPLES) FOR DIAGNOSTIC PURPOSES.
C
      INDEX=INDEX+1
   50 M=0
      DO 60 J=1,5
         ISUM(J)=0
   60 CONTINUE
   70 DO 80 J=1,5
         ISUM(J)=ISUM(J)+I(INDEX+J-1)
   80 CONTINUE
      M=M+1
      IF(M-50) 90,100,100
   90 INDEX=INDEX+5
      IF(INDEX.GT.2005) GOTO 140
```

```
      GOTO 70
  100 DO 110 J=1,5
         AVRIJ(J)=FLOAT(ISUM(J))/FLOAT(M)
  110 CONTINUE
      TEMP1=SCALE(RCDSEC)
      IF(MOD(NLINES,20).EQ.0) WRITE(6,1400) NBLOCK,RCDHRS,RCDMIN,TEMP1,
     *                                   (J,J=1,5)
 1400 FORMAT('0  RECORD ',I5,'  STARTS AT TIME ',2I2,':',F7.4/
     *'0',7X,'CHANNEL',11X,5(I1,11X)/)
      WRITE(6,1500) (AVRIJ(J),J=1,5)
 1500 FORMAT(' FIFTY POINT AVERAGE:  ',5(F7.2,5X))
      NLINES=NLINES+1
      INDEX=INDEX+5
      IF(INDEX.LE.2005) GOTO 50
  120 M=0
      DO 130 J=1,5
         ISUM(J)=0
  130 CONTINUE
C
C  REFILL DATA ARRAY.
C
  140 CALL TPGETZ(12,I)
      CALL TPCHKZ(12,NBYTES,EOF)
      NBLOCK=NBLOCK+1
      CALL GETIME
      IF(RCDTIM.GT.ENDTIM) STOP
      INDEX=6
      GOTO 70
C
C  COME HERE FOR A PARITY ERROR WHILE READING A RECORD.
C  IF READING ERROR OCCURRED DURING PROCESSING
C  RESUME PROCESSING WITH NEXT RECORD.
C
  150 NBLOCK=NBLOCK+1
      WRITE(6,1600) NBLOCK,ERRCD
 1600 FORMAT('0  ERROR IN READING RECORD ',I5,'.  TAPE ERROR ',I10,'.'/)
      IF(IFLAG-2) 20,160,170
  160 NXTSEC=NXTSEC+4000
      NXTTIM=NXTTIM+4000
      GOTO 30
  170 WRITE(6,1700)
 1700 FORMAT('0  ABANDONNING ALL SUMMATIONS FOR THIS RECORD.'/
     *'0  RESUME AVERAGING WITH NEXT RECORD.'/)
      NLINES=0
      NXTSEC=NXTSEC+4000
      NXTTIM=NXTTIM+4000
      GOTO 120
C
C  COME HERE FOR AN END OF FILE ON TAPE.
C
  180 WRITE(6,1800) NBLOCK
 1800 FORMAT('0 END OF FILE ON TAPE.'/'0 LAST RECORD READ WAS ',I5,'.')
      STOP
      END
```

```
      SUBROUTINE GETIME
C
C  THIS SUBROUTINE DECODES THE THREE TIME WORDS INTO HOURS, MINUTES
C  AND SECONDS.
C
      INTEGER*2 I(1),K(16)
      INTEGER RCDHRS,RCDMIN,RCDSEC,RCDTIM,TIMSEC
      COMMON RCDTIM,NXTTIM,RCDHRS,NXTHRS,RCDMIN,NXTMIN,RCDSEC,NXTSEC,I
C
C  SET RECORD START TIME TO TIME CODE OF LAST RECORD.
C
      RCDHRS=NXTHRS
      RCDMIN=NXTMIN
      RCDSEC=NXTSEC
      RCDTIM=NXTTIM
      KWDNUM=2006
C
C  BREAK DOWN 16 BIT TIME WORD INTO SEPARATE BITS AND STORE
C  IN LENGTH 16 ARRAY "K".
C
   10 ITIME=I(KWDNUM)
      IST=1
      IF(ITIME.GE.0) GO TO 20
      K(16)=1
      ITIME=ITIME+32768
      IST=2
   20 DO 40 M=IST,16
         N=17-M
         ITEST=2**(N-1)
         IF(ITIME.GE.ITEST) GO TO 30
         K(N)=0
         GOTO 40
   30    K(N)=1
         ITIME=ITIME-ITEST
   40 CONTINUE
      IF(ITIME.NE.0) WRITE(6,1000)
 1000 FORMAT('   ERROR IN READING TIME WORD. RESULT NOT ZERO.')
      IF(KWDNUM-2007) 50,70,80
C
C  COMPUTE FRACTIONAL SECONDS IN TENTHS OF MILLISECONDS.
C
   50 MULT=1
      MSECS=0
      DO 60 IDX=1,13,4
         ITMSEC=8*K(IDX+3)+4*K(IDX+2)+2*K(IDX+1)+K(IDX)
         MSECS=MULT*ITMSEC+MSECS
         MULT=10*MULT
   60 CONTINUE
      KWDNUM=2007
      GOTO 10
C
C  COMPUTE SECONDS AND MINUTES.
C
   70 NXTSEC=40*K(7)+20*K(6)+10*K(5)+8*K(4)+4*K(3)+2*K(2)+K(1)
```

```
      NXTSEC=10000*NXTSEC+MSECS
      NXTMIN=40*K(15)+20*K(14)+10*K(13)+8*K(12)+4*K(11)+2*K(10)+K(9)
      KWDNUM=2008
      GOTO 10
C
C  COMPUTE HOURS.
C
   80 NXTHRS=20*K(6)+10*K(5)+8*K(4)+4*K(3)+2*K(2)+K(1)
C
C  CONVERT TIME TO TENTHS OF MILLISECONDS.
C
      NXTTIM=TIMSEC(NXTHRS,NXTMIN,0.0)+NXTSEC
      RETURN
      END
      INTEGER FUNCTION TIMSEC(HOUR,MIN,SEC)
C
C  THIS FUNCTION CONVERTS TIME IN HOURS, MINUTES AND SECONDS
C  TO TIME IN TENTHS OF MILLISECONDS.
C
      INTEGER HOUR,TIMSEC
      REAL SEC
      TIMSEC=36000000*HOUR+600000*MIN+INT(10000.0*SEC)
      RETURN
      END
      REAL FUNCTION SCALE(INTSEC)
C
C  THIS FUNCTION CONVERTS TIME IN TENTHS OF MILLISECONDS
C  TO TIME IN SECONDS.
C
      SCALE=FLOAT(INTSEC)/10000.0
      RETURN
      END
//GO.FT12F001 DD UNIT=TAPE,VOL=(,RETAIN,SER=WI20DA),DISP=OLD,
// DCB=(RECFM=U,BLKSIZE=4016),LABEL=(1,BLP,,IN)
//GO.SYSIN DD *
1650:05.00
1651:20.00
/*
```

APPENDIX VII. Listing of CDC Fortran IV program CONVWAL.

The first nineteen lines of this appendix are a Cyber procedure file which will run program CONVWAL as a batch job. This procedure file was used to convert 67 seconds (T+29 → T+95) of differential absorption data (tape WI20DA) to Cyber format (direct access file DA1020). The direct access file is defined on line number 11; the central memory field length is set to 177000 (octal) on line 10; and the central processing unit (CPU) time limit is raised from the default value of 8 seconds to 50 seconds on line 12.

Within program CONVWAL logical variable PARERR is used as a flag to indicate a parity error in reading the current tape record. Processing skips over records with parity errors. A message is printed to the diagnostic file (TAPE2=OUTPUT) that a parity error occurred.

```
/JOB
/NOSEQ
MKM.
SIGNON(3MIKEKM)
BILL,ELEC,PS2714.
USE,OPTION,CONVWAL.
GET,OPTION.
PRINT.
GET,CONVWAL.
RFL,177000.
DEFINE,DA1020.
SETTL,50.
GRAB,UOILIB.
FTN,I=CONVWAL,L=0,A.
LABEL(TAPE,VSN=WI20DA-C383,NT,D=800,PO=UR,F=L,LB=KU)
LGO,,,TAPE,DA1020.
UNLOAD,TAPE.
/EOR
 29, 95, 5000, 16,52,00, 18.1020
```

```
      PROGRAM CONVWAL(INPUT,OUTPUT,WAL=0,CYB,TAPE1=INPUT,
     *                TAPE2=OUTPUT,TAPE3=WAL,TAPE4=CYB)
C
C  WRITTEN BY FRANK M. BRASWELL        JULY 1979
C  MODIFIED BY M K MCINERNEY     SEPTEMBER 1979
C
C  FOR THE CDC CYBER 175
C
C  THIS PROGRAM IS DESIGNED TO READ A WALLOPS ISLAND (WI)
C  DIGITAL TAPE AND CONVERT IT TO A CYBER COMPATIBLE
C  TAPE OR DISK FILE WITH REFORMATTED DATA.
C
C  ********************************************************
C
C  HOW TO USE:
```

```
C
C  THE USER MUST SUPPLY SEVERAL THINGS.
C
C       TAPE 1) INPUT FILE CONSISTING OF THE RANGE OF SECONDS
C               YOU WISH TO CONVERT,DIGITIZING RATE AND THE
C               LAUNCH TIME AND FLIGHT.
C
C               ENTER IN THE FOLLOWING ORDER:
C
C               FIRST SECOND (INTEGER)
C               LAST SECOND (INTEGER)
C               DIGITIZING RATE (INTEGER)
C               HOURS (INTEGER)
C               MINUTES (INTEGER)
C               SECONDS (REAL)
C               FLIGHT NUMBER (REAL)
C
C               IT IS FORMAT FREE INPUT SO THE VALUES NEED ONLY BE
C               SEPARATED BY COMMAS OR BLANKS.
C
C       TAPE 2) NAME OF OUTPUT FILE FOR DIAGNOSTICS.
C
C       TAPE 3) NAME OF LOCAL FILE REFERENCE OF A WI TAPE. THE
C               TAPE CONTAINS 4 HEADER RECORDS AND 5 CALIBRATION
C               RECORDS.  IF THE NUMBER OF HEADER OR CALIBRATION
C               RECORDS IS DIFFERENT CHANGE THE VALUES OF NUMCAL
C               AND NUMHEAD IN SUBROUTINE CALHED.
C
C       TAPE 4) NAME OF LOCAL FILE REFERENCE OF EITHER A TAPE OR
C               DISK FILE  WHICH WILL RECEIVE THE REFORMATTED
C               DATA.
C
C  THE CALL IS MADE WITH THE FILES IN THE FOLLOWING ORDER:
C       LGO,<TAPE1>,<TAPE2>,<TAPE3>,<TAPE4>
C
C  **********************************************************
C
C  THE WI TAPE IS MADE UP OF SEVERAL HEADER AND CALIBRATION
C  RECORDS WHICH ARE SKIPPED OVER.  THE DATA IS IN IBM HALF
C  WORD INTEGERS 16 BITS OR 2 BYTES LONG.  EACH RECORD IS
C  2008 HALFWORDS LONG.  THE FIRST 5 HALFWORDS ARE GARBAGE
C  FOLLOWED BY 2000 DATA POINTS.  THE LAST 3 HALFWORDS
C  CONTAIN THE TIME CODE WHICH IS THE TIME OF THE FIRST DATA
C  POINT OF THE FOLLOWING RECORD.
C
C       THE DATA IS READ IN AND REFORMATTED AS FOLLOWS:
C            ONE 60 BIT REAL:  TIME FROM LAUNCH
C            FIVE THOUSAND 60 BIT INTEGERS:  DATA
C
C  THE TIME FROM LAUNCH MUST FALL ON THE SECOND.
C
C  THE WI DATA IS READ INTO AN ARRAY OF 540  60 BIT CYBER WORDS.
C  THE 16 BIT INTEGERS PACKED INTO THE ARRAY ARE UNPACKED INTO
C  AN ARRAY OF 5000  60 BIT CYBER INTEGERS.
```

```
C
C  SINCE THE 16 BIT INTEGERS ARE POSITIVE THEY CAN BE
C  UNPACKED DIRECTLY INTO THE 16 LOW ORDER BITS OF THE 60 BIT
C  CYBER WORDS WITHOUT HAVING TO WORRY ABOUT THE SIGN BIT.
C
C  THE TIME OF A WI RECORD IS OBTAINED BY USING SUBROUTINES
C  RECTIM, TPGET, GETIME, AND IBIN.
C
      DIMENSION ICYBER(5000),ICHAN1(5)
      LOGICAL PARERR
      COMMON BUFFER(540)
      DATA ICHAN1/1,0,-1,-2,2/
C
C  READ IN START AND STOP TIMES, DIGITIZING RATE, LAUNCH TIME
C  AND FLIGHT NUMBER.
C
      READ(1,*) IFIRST,LAST,IDIGRAT,IHRS,MIN,SEC,FLIGHT
C
C  CONVERT START AND STOP TIMES TO SECONDS.
C
      TIMLAU=FLOAT(3600*IHRS+60*MIN)+SEC
      SEC1=FLOAT(IFIRST)+TIMLAU
      FIRST=SEC1
      SEC2=FLOAT(LAST)+TIMLAU
      WRITE(2,1000) FLIGHT
 1000 FORMAT(" THE FLIGHT NUMBER IS: ",F10.4/)
      WRITE(2,1100) TIMLAU,IHRS,MIN,SEC
 1100 FORMAT(" LAUNCH TIME IS: ",F15.5,"  ----  ",I2,":",I2,":",F7.4/)
      WRITE(2,1200) SEC1,IFIRST,SEC2,LAST
 1200 FORMAT(" FIRST SECOND WILL BE: ",F15.4,"  ----  ",I4/
     *" TERMINATING SECOND WILL BE: ",F15.4,"  ----  ",I4/)
C
C  CYBLEN IS THE LENGTH IN TIME OF 1 CYBER RECORD OF 5000 POINTS.
C  RECLEN IS THE LENGTH IN TIME OF 1 WI RECORD OF 2000 POINTS.
C  DATATI IS THE TIME BETWEEN DATA POINTS IN SECONDS.
C
      DIGRAT=FLOAT(IDIGRAT)
      CYBLEN=5000./DIGRAT
      RECLEN=2000./DIGRAT
      DATATI=1./DIGRAT
C
C  INITIALIZE RECORD COUNTER.
C
      IRECNO=0
C
C  SKIP OVER CALIBRATION AND HEADER RECORDS.
C
      CALL CALHED(IRECNO)
C
C  LOCATE BEGINNING RECORD BY FETCHING THE TIME CODE FROM THE
C  END OF EACH WALLOPS RECORD.  THE TIME CODE IS STORED IN
C  THE 535TH AND 536TH WORD OF THE CYBER INPUT BUFFER EXACTLY
C  40 BITS OFF THE 535TH WORD.
C
```

```
   10 CALL RECTIM(TIME,.FALSE.,IRECNO,PARERR)
      IF(PARERR) GOTO 10
C
C  RECALL THAT THE TIME CODE ON THIS RECORD POINTS TO THE
C  BEGINNING OF THE NEXT RECORD SO WE ARE CHECKING TO SEE IF
C  SEC1 FALLS IN THE PREVIOUS RECORD.
C
C  THIS IS THE REASON FOR THE DUMMY CALL IN THE NEXT STEP.
C
      IF(SEC1.GT.(TIME+RECLEN)) GOTO 10
C
C  DUMMY CALL TO PULL IN NEXT DATA RECORD.
C
      CALL TPGET(3,ICYBER,1,1,1,0,IRECNO,.FALSE.,PARERR)
C
C  CALCULATE WHICH DATA POINT CORRESPONDS TO THE START TIME.
C       ( THE LOCATION IN THE WI ARRAY IS CALCULATED EACH TIME
C         THRU THE CONVERSION LOOP TO SEE IF THE TIME CODE MATCHES
C         UP WITH THE DATA POINTS.)
C
   20 IFRSTRC=IRECNO
      LOC1=IFIX((SEC1-TIME)/DATATI)
C
C  DATA CONVERSION MUST BEGIN WITH CHANNEL 1.
C
      LOC1=LOC1+ICHAN1(MOD(LOC1,5)+1)
C
C  THE CALCULATED LOCATION IS USED ONLY FOR THE FIRST RECORD.
C
      IF(SEC1.EQ.FIRST) LOC=LOC1
C
C  OFFSET 5 LOCATIONS IN WI ARRAY TO AVOID THE FIRST 5
C  GARBAGE POINTS.
C
      LOC=LOC+5
C
C  CALCULATE THE NUMBER OF POINTS LEFT AT THE END OF THE
C  WI ARRAY.
C
      LEFT=(2005-LOC)+1
C
C  PRINT OUT LOC, LOC1 AND LEFT FOR DIAGNOSTIC PURPOSES.
C
      WRITE(2,1300) LOC,LEFT,LOC1
 1300 FORMAT(" LOCATION TO START ",I5," # LEFT IN WALLARRAY: ",
     * I5," CALCULATED LOCATION TO START: ",I5)
C
C  THE INPUT BUFFER IS A CYBER ARRAY WHICH HOLDS THE 2008
C  16 BIT INTEGERS OF THE WI DATA RECORDS.
C
C  COMPUTE THE NUMBER OF BITS OF DATA.
C
      NUMBITS=16*(LOC-1)
C
```

```
C  LOCATE THE WORD AND BIT IN THE CYBER ARRAY CORRESPONDING
C  TO AN ARRAY LOCATION IN THE WI ARRAY OF 2008.
C
      IWORDS=NUMBITS/60+1
      IBITS=MOD(NUMBITS,60)
C
C  PICK OFF DATA POINTS ON FIRST RECORD.
C
      IF(.NOT.PARERR) CALL TPGET(3,ICYBER,1,LEFT,IWORDS,IBITS,IRECNO,
     *.TRUE.,PARERR)
      IF(PARERR) CALL FILL(ICYBER,1,LEFT)
C
C  FILL IN ANOTHER 2000 WORDS OF THE ICYBER ARRAY AND
C  INCREMENT INDEX.
C
      INDCYB=LEFT+1
      CALL TPGET(3,ICYBER,INDCYB,2000,2,20,IRECNO,.FALSE.,PARERR)
      IF(PARERR) CALL FILL(ICYBER,INDCYB,2000)
      INDCYB=INDCYB+2000
C
C  DETERMINE IF ARRAY ICYBER WILL HOLD ANOTHER 2000 DATA POINTS.
C
      IF((5000-INDCYB).LT.2000) GOTO 30
C
C  FILL IN ANOTHER 2000 WORDS OF THE ICYBER ARRAY AND
C  INCREMENT INDEX.
C
      CALL TPGET(3,ICYBER,INDCYB,2000,2,20,IRECNO,.FALSE.,PARERR)
      IF(PARERR) CALL FILL(ICYBER,INDCYB,2000)
      INDCYB=INDCYB+2000
C
C  FILL IN THE LAST DATA POINTS OF A CYBER RECORD.
C
   30 LOC=5000-INDCYB+1
C
C  OBTAIN THE TIME CODE OF THE CURRENT RECORD.
C
      CALL RECTIM(TIME,.TRUE.,IRECNO,PARERR)
      IF(PARERR) TIME=TIME+RECLEN*FLOAT(IRECNO-IFRSTRC)
      CALL TPGET(3,ICYBER,INDCYB,LOC,2,20,IRECNO,.FALSE.,PARERR)
      IF(PARERR) CALL FILL(ICYBER,INDCYB,LOC)
C
C  OUTPUT A CYBER RECORD OF 5000 POINTS AND THE TIME
C  AFTER LAUNCH.
C
      WRITE(4) (SEC1-TIMLAU),ICYBER
C
C  PREPARE THE TIME OF THE NEXT RECORD.
C
      SEC1=SEC1+CYBLEN
      LOC=LOC+1
C
C  OUTPUT TIME, SEC1 AND TIME+RECLEN FOR DIGNOSTIC PURPOSES.
C
```

```
      WRITE(2,1400) TIME,SEC1,(TIME+RECLEN)
 1400 FORMAT(" THE TIME ON THIS WAL REC IS: ",F15.5,
     * "  SECOND OF NEXT CYBER REC IS: ",F15.5,/,
     * "  THE TIME OF THE NEXT WAL REC IS: ",F15.5)
C
C  CHECK FOR THE END.
C
      IF(SEC1.GT.SEC2) GOTO 40
C
C  OTHERWISE CONTINUE FILLING CYBER RECORDS.
C
      GOTO 20
   40 WRITE(2,1500) IRECNO
 1500 FORMAT("0CONVERSION FINISHED ON REC: ",I7)
      STOP
      END
      SUBROUTINE FILL(IARRAY,IFIRST,NUMBER)
C
C  THIS SUBROUTINE FILLS IARRAY WITH THE INTEGER VALUE OF 9000
C  BEGINNING WITH INDEX IFIRST AND FOR NUMBER ARRAY LOCATIONS
C  AFTER IFIRST.
C  THE INTEGER 9000 IS USED AS A FILLER BECAUSE THE DIGITIZER
C  HAS A RESOLUTION OF 12 BITS WHICH CORRESPONDS TO A MAXIMUM
C  INTEGER VALUE OF 4095.
C
      DIMENSION IARRAY(1)
      DO 10 I=1,NUMBER
         IARRAY(IFIRST+I-1)=9000
   10 CONTINUE
      RETURN
      END
      SUBROUTINE RECTIM(TIME,AHEAD,IRECNO,PARERR)
C
C  THIS ROUTINE FETCHES THE 3 TIME WORDS OFF THE WI RECORD
C  AND DECODES THEM USING THE GETIME ROUTINE.
C
C  VARIABLE AHEAD MOVES AHEAD ONE RECORD AT A TIME
C  OR FETCHS THE TIME OFF THE CURRENT RECORD WITHOUT
C  MOVING AHEAD.
C
      DIMENSION ITIM(3)
      LOGICAL AHEAD,PARERR
      COMMON BUFFER(540)
      CALL TPGET(3,ITIM,1,3,535,40,IRECNO,AHEAD,PARERR)
      IF(.NOT.PARERR) TIME=GETIME(ITIM(1),ITIM(2),ITIM(3))
      RETURN
      END
      SUBROUTINE CALHED(NBLOCK)
C
C  THIS ROUTINE SKIPS OVER THE CALIBRATION AND HEADER RECORDS.
C
C  NUMHED AND NUMCAL MAY BE CHANGED IF THE NUMBER OF HEADER
C  RECORDS IS NOT 4 AND THE NUMBER OF CALIBRATION RECORDS
C  IS NOT 5.
```

```
C
      NUMHED=4
      NUMCAL=350
      DO 10 L=1,NUMHED
         CALL FORWRD(3,NBLOCK)
         WRITE(2,1000)
 1000    FORMAT (1X,"HEADER RECORD READ")
   10 CONTINUE
      DO 20 L=1,NUMCAL
         CALL FORWRD(3,NBLOCK)
         WRITE(2,1100)
 1100    FORMAT (1X,"CAL RECORD READ")
   20 CONTINUE
      RETURN
      END
      FUNCTION GETIME(TIM1,TIM2,TIM3)
C
C  THIS ROUTINE DECODES THE TIME CODES ON THE WALLOPS
C  RECORDS AND CONVERTS HOURS, MINUTES AND SECONDS TO
C  SECONDS.
C
C  THE 3 VARIABLES CONTAINING THE WI TIME CODE ARE SENT AS
C  TIM1, TIM2, AND TIM3.  THE 16 BIT REPRESENTATIONS ARE
C  STORED IN ARRAY K AFTER THE CALLS TO IBIN AND THEN THE WEIGHTS
C  ARE MULITPLIED OUT ACCORDING TO THE CODE FOR THE
C  HOURS, MINUTES, AND SECONDS.
C
C       SEE AERONOMY REPORT #85 BY B.E. GILCHRIST;
C            TABLE ON PAGE 48 FOR DETAILS ON THE WEIGHTS.
C
      INTEGER K(16),RECHRS,RECMIN,RECSEC,TIM1,TIM2,TIM3
      CALL IBIN(K,TIM1)
      FRACS=(8*K(4)+4*K(3)+2*K(2)+K(1))*0.0001
     *+(8*K(8)+4*K(7)+2*K(6)+K(5))*0.001
     *+(8*K(12)+4*K(11)+2*K(10)+K(9))*0.01
     *+(8*K(16)+4*K(15)+2*K(14)+K(13))*0.1
      CALL IBIN(K,TIM2)
      RECSEC=40*K(7)+20*K(6)+10*K(5)+8*K(4)+4*K(3)+2*K(2)+K(1)
      RECMIN=40*K(15)+20*K(14)+10*K(13)+8*K(12)+4*K(11)+2*K(10)+K(9)
      CALL IBIN(K,TIM3)
      RECHRS=20*K(6)+10*K(5)+8*K(4)+4*K(3)+2*K(2)+K(1)
      GETIME=FLOAT(3600*RECHRS+60*RECMIN+RECSEC)+FRACS
      RETURN
      END
      SUBROUTINE IBIN(K,NUM)
C
C  THIS ROUTINE TAKES THE WORD NUM AND FILLS ARRAY K WITH
C  ONES AND ZEROS CORRESPONDING TO THE VALUES OF NUM'S
C  16 LOW ORDER BITS.  K(16) IS THE MOST SIGNIFICANT OF THE
C  SIXTEEN BITS.
C
C  THE DO LOOP HAS BEEN UNROLLED TO IMPROVE EFFICIENCY.
C
      INTEGER K(16)
```

```
      DO 10 I=1,16,4
         K(I)=MOD(NUM,2)
         NUM=NUM/2
         K(I+1)=MOD(NUM,2)
         NUM=NUM/2
         K(I+2)=MOD(NUM,2)
         NUM=NUM/2
         K(I+3)=MOD(NUM,2)
         NUM=NUM/2
   10 CONTINUE
      IF(NUM.EQ.0) RETURN
      WRITE(2,1000)
 1000 FORMAT(" "/"0*** ERROR IN TIME CODE WORD.  THE RESULT IS NOT ZERO.
     * ***")
      RETURN
      END
      SUBROUTINE TPGET(U,ARRAY,INDARR,NLOOP,INBUF,IOFSET,NBLOCK,NOREAD,
     *                 PARERR)
C
C  THIS ROUTINE ACCESSES THE DATA IN A RECORD.
C
C  WE MAY ACCESS EITHER THE CURRENT RECORD OR READ IN AND
C  ACCESS DATA ON THE NEXT RECORD DEPENDING ON THE
C  VALUE OF NOREAD.
C
      IMPLICIT INTEGER(A-Z)
      LOGICAL NOREAD,PARERR
      DIMENSION  ARRAY(1)
      COMMON BUFFER(540)
      IF(NOREAD) GOTO 20
      PARERR=.FALSE.
      BUFFER IN(U,1) (BUFFER(1),BUFFER(540))
      NBLOCK=NBLOCK+1
      IF(UNIT(U)) 10,30,40
   10 CALL LENGTHX(U,I,J)
C
C  I IS THE NUMBER OF 60-BIT WORDS READ.
C  J IS THE NUMBER OF BITS IN THE LAST 60-BIT WORD THAT WERE NOT USED.
C  PULL APART WALLOPS RECORD.
C  PLACE EACH 16 BITS (WALLOPS WORD) OF DATA INTO ONE 60 BIT
C  (CYBER WORD) WORD.
C
   20 CALL GBYTES(BUFFER(INBUF),ARRAY(INDARR),IOFSET,16,0,NLOOP)
      RETURN
   30 NBLOCK=NBLOCK-1
      WRITE(2,1000) NBLOCK
 1000 FORMAT(" "/"0","*** END OF FILE ENCOUNTERED. LAST RECORD IS ",I4,
     *". ***")
      STOP 1
   40 WRITE(2,1100) NBLOCK
 1100 FORMAT(" "/"0","*** PARITY ERROR DETECTED IN RECORD ",I4,". ***")
      PARERR=.TRUE.
      RETURN
      END
```

```
      SUBROUTINE FORWRD(U,NBLOCK)
C
C  THIS SUBROUTINE IS USED TO FORWARD SPACE ONE RECORD ON TAPE UNIT U.
C  THE NUMBER OF THE RECORD SKIPPED IS NBLOCK.
C
      IMPLICIT INTEGER(A-Z)
      DIMENSION BUFFER(540)
      BUFFER IN (U,1)(BUFFER(1),BUFFER(540))
      NBLOCK=NBLOCK+1
      IF(UNIT(U)) 10,20,30
   10 RETURN
   20 NBLOCK=NBLOCK-1
      WRITE(2,1000) NBLOCK
 1000 FORMAT(" "/"0","*** END OF FILE ENCOUNTERED. LAST RECORD IS ",I4,
     *". ***")
      STOP 2
   30 WRITE(2,1100) NBLOCK
 1100 FORMAT(" "/"0","*** PARITY ERROR DETECTED IN RECORD ",I4,". ***")
      RETURN
      END
```

APPENDIX VIII. Listings of CDC Fortran IV programs DAMED and DAAVG.

A procedure file which will run these programs as batch jobs on the Cyber is included as the first 25 lines of this appendix. By substituting DAAVG for DAMED in the procedure file, program DAAVG can be run. The attach statement on line 10 makes direct access file DA1020 (created by program CONVWAL) local. Input parameters follow the /EOR statement. Fifty seconds (1652:40UT → 1653:30UT) of differential absorption data fram channel 5 are processed. The last six input parameters are the dB calibration levels from program DACAL.

If a bad data value is encountered in the differential absorption data file, subroutine BADDATA is called. This subroutine prevents the bad data value from being processed (through setting GOODFLG false). Subroutine BADDATA also increments the bad data point counter, NBADPTS, by one.

```
/JOB
/NOSEQ
MKM.
SIGNON(3MIKEKM).
BILL,ELEC,PS2714.
USE,OPTION,DAMED,DA1020.
GET,OPTION.
PRINT/FETCH.
GET,DAMED.
ATTACH,DA1020.
FTN,I=DAMED,L=0,A.
LGO,,,OUT,DA1020.
PRINT,OUT.
/EOR
     18.1020 ORDINARY DIFFERENTIAL ABSORPTION RATES FOR 5MHZ
5
1652:00.000
1652:40.000
1653:30.000
0737.01
1374.65
2005.98
2639.50
3266.12
3910.03
```

```
      PROGRAM DAMED(INPUT,OUTPUT,DATAOT,TP,TAPE1=INPUT,TAPE2=OUTPUT,
     *              TAPE3=DATAOT,TAPE4=TP)
C
C  THIS PROGRAM COMPUTES DIFFERENTIAL ABSORPTION RATES FROM
C  DATA TAPES DIGITIZED AT THE WALLOPS PCM FACILITY.
C
C  DIGITIZED DATA MUST FIRST HAVE BEEN CONVERTED TO CYBER
C  FORMAT VIA PROGRAM CONVWAL.
C
C  THIS PROGRAM FINDS THE MEDIAN DIGITAL VALUE OVER ONE
```

```
C  SECOND INTERVALS.
C
C  THE TWO INPUT FILES ARE INPUT (TAPE1) AND TP (TAPE4).
C  FILE DATAIN CONTAINS THE INFORMATION CONCERNING THE
C  PROCESSING TO BE DONE.
C  FILE TP IS THE DIGITIZED DATA FILE (EITHER TAPE OR
C  DISK).
C
C  THE OUTPUT FILE IS DATAOT (TAPE3).
C
C  FILE DATAIN CONTAINS THE FOLLOWING 11 LINES OF INFORMATION
C  ENTERED IN THE FOLLOWING ORDER AND IN THE STATED FORMATS.
C
C       1.)  HEADING FOR OUPUT FILE.
C                 (UP TO 80 CHARACTERS ON ONE LINE)(8A10)
C       2.)  THE NUMBER OF THE CHANNEL TO BE PROCESSED.
C                 (INTEGER IN COLUMN 1)(I1)
C       3.)  THE LAUNCH TIME.
C                 (HHMM:SECS)(2I2,1X,F6.3)
C       4.)  THE TIME TO BEGIN PROCESSING.
C                 (HHMM:SECS)(2I2,1X,F6.3)
C       5.)  THE TIME TO END PROCESSING.
C                 (HHMM:SECS)(2I2,1X,F6.3)
C       6.)  THE SIX CALIBRATION LEVELS (FROM PROGRAM DACAL).
C                 (SIX LINES: BEGINNING WITH THE -50 DB LEVEL
C                             AND ENDING WITH THE  0 DB LEVEL)
C                             (6(F7.2/))
C
C  REVISED BY M K MCINERNEY   OCTOBER,1979.
C
      REAL ENDTIM,LAUNCH,ORD,RCDSEC,SEC,SRCHTM,STARTM,TDELT,VOLTS(6)
      LOGICAL ENDFLG,GOODFLG,OPENFLG,SUMFLG
      INTEGER DATA(5000),HEADNG(8),HOUR,INDEX,MIN,NBADPTS,NSMPLS,
     *XCHANL,XINDEX,XPWR(1000)
      COMMON/BAD/GOODFLG,NBADPTS
      COMMON/TIME/HOUR,MIN,SEC
      COMMON/DARATE/ORD,SUMFLG,TDELT,XPWR
C
C  INITIALIZE VARIABLES.
C
      TDELT=1.0
      RDCSEC=0.0
      ENDFLG=.FALSE.
      SUMFLG=.FALSE.
      GOODFLG=.TRUE.
      OPENFLG=.FALSE.
      NBADPTS=0
      REWIND 3
      REWIND 4
C
C  READ HEADING AND CHANNEL INFORMATION.
C  OUTPUT HEADING.
C
      READ(1,1000) (HEADNG(I),I=1,8)
```

```
 1000 FORMAT(8A10)
      READ(1,1100) XCHANL
 1100 FORMAT(I1)
      WRITE(3,1200) (HEADNG(I),I=1,8)
 1200 FORMAT("1",5X,8A10)
C
C  READ LAUNCH TIME AND PRINT OUT.
C
      CALL READTM(LAUNCH,1)
      WRITE(3,1300) HOUR,MIN,SEC
 1300 FORMAT("0",5X,"LAUNCH TIME IS ",2I2,":",F6.3)
C
C  READ START TIME AND PRINT OUT.
C
      CALL READTM(STARTM,1)
      WRITE(3,1400) HOUR,MIN,SEC
 1400 FORMAT("0",5X,2I2,":",F6.3," IS THE TIME TO BEGIN PROCESSING.")
C
C  PRINT OUT NUMBER OF CHANNEL TO BE PROCESSED.
C
      WRITE(3,1500) XCHANL
 1500 FORMAT("0     CHANNEL ",I1," WILL BE PROCESSED.")
C
C  READ STOP TIME AND PRINT OUT.
C
      CALL READTM(ENDTIM,1)
      WRITE(3,1600) HOUR,MIN,SEC
 1600 FORMAT("0",5X,2I2,":",F6.3," IS THE TIME TO END PROCESSING.")
C
C  READ CALIBRATION LEVELS AND PRINT OUT.
C
C  USE A SEPARATE HEADING FOR ORDINARY AND EXTRA-ORDINARY PROCESSING.
C
      READ(1,1700) (VOLTS(I),I=1,6)
 1700 FORMAT(6(F7.2/))
      IF(XCHANL.GE.4) GOTO 10
C
C  PRINT CALIBRATION LEVELS AND HEADING FOR EXTRA-ORDINARY
C  PROCESSING.
C
      WRITE(3,1800) (VOLTS(I),I=1,6)
 1800 FORMAT("0",5X,"EXTRA-ORDINARY CALIBRATION LEVELS:"/
     *"0",19X,"-50",2X,F7.2/" ",19X,"-40",2X,F7.2/
     *" ",19X,"-30",2X,F7.2/" ",19X,"-20",2X,F7.2/
     *" ",19X,"-10",2X,F7.2/" ",19X,"  0",2X,F7.2)
      WRITE(3,1900)
 1900 FORMAT("1",5X,"SECONDS FROM LAUNCH",5X,"MEDIAN",6X,"D.A. RATE"/
     *" ",26X,"DIGITAL VALUE",13X,"EX-ORD LEVEL"/)
      GOTO 20
C
C  PRINT CALIBRATION LEVELS AND HEADING FOR ORDINARY
C  PROCESSING.
C
   10 WRITE(3,2000) (VOLTS(I),I=1,6)
```

```
 2000 FORMAT("0",5X,"ORDINARY CALIBRATION LEVELS:"/
     *"0",13X,"-50",2X,F7.2/" ",13X,"-40",2X,F7.2/
     *" ",13X,"-30",2X,F7.2/" ",13X,"-20",2X,F7.2/
     *" ",13X,"-10",2X,F7.2/" ",13X,"  0",2X,F7.2)
      WRITE(3,2100)
 2100 FORMAT("1",5X,"SECONDS FROM LAUNCH",5X,"MEDIAN",6X,"D.A. RATE"/
     *" ",26X,"DIGITAL VALUE",15X,"ORD LEVEL"/)
C
C  CONVERT START AND END TIMES TO SECONDS FROM LAUNCH.
C  LOCATE RECORD IN WHICH PROCESSING IS TO BEGIN.
C
   20 IF(STARTM.GE.ENDTIM) GOTO 110
      ENDTIM=ENDTIM-LAUNCH
      STARTM=STARTM-LAUNCH
      SRCHTM=STARTM-0.5
   30 READ(4) RCDSEC,(DATA(I),I=1,5000)
      IF(EOF(4).NE.0.0) GOTO 80
      IF(SRCHTM.LT.RCDSEC.AND.(.NOT.OPENFLG)) GOTO 90
      OPENFLG=.TRUE.
      IF(SRCHTM.GT.(RCDSEC+1.0)) GOTO 30
C
C  CALCULATE LAST ELEMENT OF THE FRAME IMMEDIATELY PRECEDING
C  THE FRAME WHICH CONTAINS THE ARRAY ELEMENT CORRESPONDING
C  TO THE SEARCH TIME (START TIME MINUS .5 SECOND).
C
   40 CALL FINDPT(SRCHTM,RCDSEC,INDEX)
C
C  FILL ARRAY XPWR WITH ONE SECOND (1000 POINTS) OF DATA.
C  IF A DATA VALUE IS LESS THEN ZERO OR GREATER THAN 4095
C  (12 BIT DIGITIZER RESOLUTION) THEN FLAG THAT POINT AS
C  BAD DATA AND LEAVE OUT OF MEDIAN SEARCH.
C
      NSMPLS=1
   50 IF(INDEX.GE.5000) GOTO 60
      XINDEX=INDEX+XCHANL
      IF(DATA(XINDEX).LT.0.OR.DATA(XINDEX).GT.4095) CALL BADDATA
      IF(GOODFLG) XPWR(NSMPLS-NBADPTS)=DATA(XINDEX)
      INDEX=INDEX+5
      NSMPLS=NSMPLS+1
      GOODFLG=.TRUE.
      IF(NSMPLS.GT.1000) GOTO 70
      GOTO 50
C
C  REFILL ARRAY DATA WITH ONE SECOND OF DATA.
C
   60 READ(4) RCDSEC,(DATA(I),I=1,5000)
      IF(EOF(4).NE.0.0) GOTO 100
      IF(RCDSEC.GE.ENDTIM) ENDFLG=.TRUE.
      INDEX=0
      GOTO 50
C
C  FIND MEDIAN VALUE OF DATA.
C
   70 CALL FINMED((NSMPLS-1-NBADPTS),STARTM,VOLTS)
```

```
      IF(ENDFLG) STOP
      NSMPLS=1
      GOTO 50
C
C  COME HERE IF AN END OF FILE IS REACHED ON THE DATA FILE
C  BEFORE THE START TIME IS FOUND.
C
   80 WRITE(3,2200)
 2200 FORMAT("0",5X,"END OF FILE ON TAPE BEFORE START TIME REACHED.")
      STOP1
C
C  COME HERE IF THE START TIME IS LESS THAN THE TIME OF THE
C  FIRST DATA RECORD.
C
   90 WRITE(3,2300)
 2300 FORMAT("0",5X,"START TIME LESS THAN FIRST DATA RECORD TIME ON TAPE
     *.")
      OPENFLG=.TRUE.
      STARTM=RCDSEC+1.0
      SRCHTM=RCDSEC+0.5
      GOTO 40
C
C  COME HERE IF AN END OF FILE WAS REACHED ON THE DATA FILE
C  BEFORE THE STOP TIME WAS FOUND.
C
  100 WRITE(3,2400) RCDSEC
 2400 FORMAT("0",5X,"END OF FILE ON TAPE BEFORE END TIME REACHED."/
     *" ",5X,"TIME OF LAST DATA RECORD READ WAS",F6.1,".")
      STOP2
C
C  COME HERE IF THE START TIME IS GREATER THEN OR EQUAL TO
C  THE STOP TIME.
C
  110 WRITE(3,2500)
 2500 FORMAT("0",5X,"STARTING TIME IS GREATER THAN OR EQUAL TO ENDING TI
     *ME."/"0",5X,"EXECUTION IS TERMINATED.")
      STOP3
      END
      SUBROUTINE FINMED(ISTOP,TSS,VOLTS)
C  THIS SUBROUTINE FINDS THE MEDIAN VALUE OF ARRAY XPWR.
C
C  A VALUE IS SELECTED FROM ARRAY XPWR FOR COMPARISON.
C  ALL ELEMENTS OF ARRAY XPWR ARE COMPARED TO THE TEST
C  VALUE AND AN ARRAY OF XPWR VALUES WHICH ARE LESS THAN
C  THE TEST VALUE IS ASSEMBLED ALONG WITH AN ARRAY OF
C  XPWR VALUES WHICH ARE GREATER THAN THE TEST VALUE.
C  THE TOTAL NUMBER OF VALUES WHICH ARE LESS THAN THE TEST
C  VALUE IS COMPARED TO THE TOTAL NUMBER OF VALUES WHICH
C  ARE GREATER THAN THE TEST VALUE.  IF THIS COMPARISON
C  RESULTS IN A TOTAL NUMBER DIFFERENCE GREATER THAN ONE
C  THEN ARRAY XPWR IS FILLED WITH THE VALUES OF THE ARRAY
C  (CONTAINING EITHER GREATER THAN OR LESS THAN VALUES)
C  IN WHICH THE MEDIAN VALUE WILL BE FOUND.  A VALUE IS
C  SELECTED FROM XPWR FOR COMPARISON AND THE ITERATION
```

```
C  PROCEEDS AS ABOVE.
C
C        NLT - NUMBER LESS THAN
C        NGT - NUMBER GREATER THAN
C        NEQ - NUMBER EQUAL TO
C       PNLT - PREVIOUS NUMBER LESS THAN
C       PNGT - PREVIOUS NUMBER GREATER THAN
C       TNLT - TOTAL NUMBER LESS THAN
C       TNGT - TOTAL NUMBER GREATER THAN
C
      REAL HALF,ORD,TDELT,VOLTS(6),X,Y
      LOGICAL GOODFLG,SUMFLG
      INTEGER ISTOP,ITEST,NEQ,NGT,NLT,PNGT,PNLT,TNGT,TNLT,XPGT(1000),
     *XPLT(1000),XPWR(1000)
      COMMON/BAD/GOODFLG,NBADPTS
      COMMON/DARATE/ORD,SUMFLG,TDELT,XPWR
C
C  SET TERMINATION LEVEL AND INITIALIZE COUNTERS.
C
      HALF=FLOAT(ISTOP)/2.0
      PNGT=0
      PNLT=0
C
C     ---   START OF ITERATION LOOP   ---
C
C  CHOOSE ELEMENT IN MIDDLE OF ARRAY TO USE AS TEST VALUE.
C
   10 ITEST=XPWR((ISTOP+1)/2)
C
C  INITIALIZE COUNTERS.
C
      NLT=0
      NGT=0
      NEQ=0
C
C  DIVIDE ARRAY XPWR INTO LESS THAN AND GREATER THAN
C  ARRAYS.
C
      DO 50 K=1,ISTOP
         IF(XPWR(K)-ITEST) 20,30,40
   20    NLT=NLT+1
         XPLT(NLT)=XPWR(K)
         GOTO 50
   30    NEQ=NEQ+1
         GOTO 50
   40    NGT=NGT+1
         XPGT(NGT)=XPWR(K)
   50 CONTINUE
C
C  COMPUTE TOTAL NUMBER LESS THAN AND TOTAL NUMBER
C  GREATER THAN.
C  TEST FOR TERMINATION CONDITION.
C
      TNLT=PNLT+NLT
```

```
      TNGT=PNGT+NGT
      X=FLOAT(TNGT+NEQ)
      Y=FLOAT(TNLT+NEQ)
      IF(X.GE.HALF.AND.Y.GE.HALF) GOTO 90
C
C  IF TERMINATION CONDITION IS NOT MET, THEN FILL ARRAY
C  XPWR WITH THE PART (LESS THAN OR GREATER THAN) OF ARRAY
C  IN WHICH THE MEDIAN VALUE WILL BE FOUND.
C
      IF(TNGT.GT.TNLT) GOTO 70
      ISTOP=NLT
      PNGT=NGT+NEQ+PNGT
      DO 60 K=1,ISTOP
         XPWR(K)=XPLT(K)
   60 CONTINUE
      GOTO 10
   70 ISTOP=NGT
      PNLT=NLT+NEQ+PNLT
      DO 80 K=1,ISTOP
         XPWR(K)=XPGT(K)
   80 CONTINUE
      GOTO 10
C
C  CONVERT MEDIAN DATA VALUE TO DB POWER LEVEL.
C
   90 CALL CONVRT(FLOAT(ITEST),TSS,VOLTS)
      RETURN
      END
      SUBROUTINE CONVRT(TEST,TSS,VOLTS)
C
C  THIS SUBROUTINE CONVERTS DATA VALUE TEST TO A DB POWER
C  LEVEL VOLTS.
C
      REAL ORD,TDELT,TEST,VOLTS(6)
      LOGICAL GOODFLG,SUMFLG
      INTEGER NBADPTS,XPWR(1000)
      COMMON/BAD/GOODFLG,NBADPTS
      COMMON/DARATE/ORD,SUMFLG,TDELT,XPWR
C
C  LOCATE DB LEVELS WHICH BRACKET THE DATA VALUE.
C
      I=1
      IF(TEST.LT.VOLTS(I)) GOTO 30
   10 I=I+1
      IF(TEST.LT.VOLTS(I)) GOTO 20
      IF(I.NE.6) GOTO 10
      EXORD=0.
      GOTO 40
C
C  COMPUTE INTERPOLATED DB LEVEL.
C
   20 X=10.*(TEST-VOLTS(I))/(VOLTS(I-1)-VOLTS(I))
      EXORD=-60.+FLOAT(10*I)-X
      GOTO 40
```

```
   30 EXORD=-50.
   40 IF(SUMFLG) GOTO 50
C
C  IF FIRST TIME THROUGH, A POWER DIFFERENCE CANNOT BE
C  COMPUTED.  THEREFORE PRINT OUT *'S.
C
      SUMFLG=.TRUE.
      ORD=EXORD
      WRITE(3,1000) TSS
 1000 FORMAT(13X,F6.2,25X,"*.***")
      GOTO 60
C
C  COMPUTE DB DIFFERENCE.
C
   50 EXXOR=EXORD
      EXORD=EXXOR-ORD
C
C  PRINT OUT TIME AND DB DIFFERENCE.
C
      WRITE(3,1100) TSS,EXORD
 1100 FORMAT(13X,F6.2,23X,F7.3)
      ORD=EXXOR
   60 WRITE(3,1200) TEST,ORD
 1200 FORMAT(29X,F8.1,18X,F8.3/)
C
C  INCREMENT TIME FOR NEXT PASS.
C
      TSS=TSS+TDELT
C
C  PRINT OUT NUMBER OF BAD DATA POINTS WHICH OCCURRED DURING
C  THE LAST SECOND.
C
      IF(NBADPTS.EQ.0) RETURN
      WRITE(3,1300) NBADPTS
 1300 FORMAT("+",55X,"THERE WERE",I3," BAD DATA POINTS DURING THE LAST S
     *ECOND.")
      NBADPTS=0
      RETURN
      END
      SUBROUTINE FINDPT(SRCHTM,RCDTIM,POINT)
C
C  THIS SUBROUTINE LOCATES THE LAST ARRAY ELEMENT OF THE FRAME
C  WHICH PRECEDES THE FRAME THAT CONTAINS THE ARRAY ELEMENT
C  CORRESPONDING TO THE SEARCH TIME.
C
      REAL RCDTIM,SRCHTM
      INTEGER POINT
      POINT=INT((SRCHTM-RCDTIM)*5000.0)
      POINT=POINT-MOD(POINT,5)
      RETURN
      END
      SUBROUTINE READTM(TIME,UNIT)
C
C  THIS SUBROUTINE READS IN TIME VALUES ENCODED IN HOURS,
```

```
C  MINUTES AND SECONDS AND CONVERTS THIS TIME TO SECONDS.
C
      REAL SEC,TIME
      INTEGER HOUR,MIN,UNIT
      COMMON/TIME/HOUR,MIN,SEC
      READ(UNIT,1000) HOUR,MIN,SEC
 1000 FORMAT(2I2,1X,F6.3)
      TIME=3600.0*FLOAT(HOUR)+60.0*FLOAT(MIN)+SEC
      RETURN
      END
      SUBROUTINE BADDATA
C
C  THIS SUBROUTINE KEEPS TRACK OF THE NUMBER OF BAD DATA VALUES
C  (<0 OR >4095).
C
      INTEGER NBADPTS
      LOGICAL GOODFLG
      COMMON/BAD/GOODFLG,NBADPTS
      NBADPTS=NBADPTS+1
      GOODFLG=.FALSE.
      RETURN
      END
```

```
      PROGRAM DAAVG(INPUT,OUTPUT,DATAOT,TP,TAPE1=INPUT,TAPE2=OUTPUT,
     *             TAPE3=DATAOT,TAPE4=TP)
C
C  THIS PROGRAM COMPUTES DIFFERENTIAL ABSORPTION RATES FROM
C  DATA TAPES DIGITIZED AT THE WALLOPS PCM FACILITY.
C
C  DIGITIZED DATA MUST FIRST HAVE BEEN CONVERTED TO CYBER
C  FORMAT VIA PROGRAM CONVWAL.
C
C  THIS PROGRAM COMPUTES THE AVERAGE DIGITAL VALUE OVER ONE
C  SECOND INTERVALS.
C
C  THE TWO INPUT FILES ARE INPUT (TAPE1) AND TP (TAPE4).
C  FILE DATAIN CONTAINS THE INFORMATION CONCERNING THE
C  PROCESSING TO BE DONE.
C  FILE TP IS THE DIGITIZED DATA FILE (EITHER TAPE OR
C  DISK).
C
C  THE OUTPUT FILE IS DATAOT (TAPE3).
C
C  FILE DATAIN CONTAINS THE FOLLOWING 11 LINES OF INFORMATION
C  ENTERED IN THE FOLLOWING ORDER AND IN THE STATED FORMATS.
C
C       1.)  HEADING FOR OUPUT FILE.
C                (UP TO 80 CHARACTERS ON ONE LINE)(8A10)
C       2.)  THE NUMBER OF THE CHANNEL TO BE PROCESSED.
C                (INTEGER IN COLUMN 1)(I1)
C       3.)  THE LAUNCH TIME.
C                (HHMM:SECS)(2I2,1X,F6.3)
C       4.)  THE TIME TO BEGIN PROCESSING.
C                (HHMM:SECS)(2I2,1X,F6.3)
C       5.)  THE TIME TO END PROCESSING.
C                (HHMM:SECS)(2I2,1X,F6.3)
C       6.)  THE SIX CALIBRATION LEVELS (FROM PROGRAM DACAL).
C                (SIX LINES: BEGINNING WITH THE -50 DB LEVEL
C                            AND ENDING WITH THE  0 DB LEVEL)
C                            (6(F7.2/))
C
C  REVISED BY M K MCINERNEY   OCTOBER,1979.
C
      REAL ENDTIM,LAUNCH,ORD,RCDSEC,SEC,SRCHTM,STARTM,TDELT,VOLTS(6)
      LOGICAL ENDFLG,GOODFLG,OPENFLG,SUMFLG
      INTEGER DATA(5000),HEADNG(8),HOUR,INDEX,MIN,NBADPTS,NSMPLS,
     *XCHANL,XINDEX,XPWR(1000)
      COMMON/BAD/GOODFLG,NBADPTS
      COMMON/TIME/HOUR,MIN,SEC
      COMMON/DARATE/ORD,SUMFLG,TDELT,XPWR
C
C  INITIALIZE VARIABLES.
C
      TDELT=1.0
      RDCSEC=0.0
      ENDFLG=.FALSE.
      SUMFLG=.FALSE.
```

```
      GOODFLG=.TRUE.
      OPENFLG=.FALSE.
      NBADPTS=0
      REWIND 3
      REWIND 4
C
C  READ HEADING AND CHANNEL INFORMATION.
C  OUTPUT HEADING.
C
      READ(1,1000) (HEADNG(I),I=1,8)
 1000 FORMAT(8A10)
      READ(1,1100) XCHANL
 1100 FORMAT(I1)
      WRITE(3,1200) (HEADNG(I),I=1,8)
 1200 FORMAT("1",5X,8A10)
C
C  READ LAUNCH TIME AND PRINT OUT.
C
      CALL READTM(LAUNCH,1)
      WRITE(3,1300) HOUR,MIN,SEC
 1300 FORMAT("0",5X,"LAUNCH TIME IS ",2I2,":",F6.3)
C
C  READ START TIME AND PRINT OUT.
C
      CALL READTM(STARTM,1)
      WRITE(3,1400) HOUR,MIN,SEC
 1400 FORMAT("0",5X,2I2,":",F6.3," IS THE TIME TO BEGIN PROCESSING.")
C
C  PRINT OUT NUMBER OF CHANNEL TO BE PROCESSED.
C
      WRITE(3,1500) XCHANL
 1500 FORMAT("0     CHANNEL ",I1," WILL BE PROCESSED.")
C
C  READ STOP TIME AND PRINT OUT.
C
      CALL READTM(ENDTIM,1)
      WRITE(3,1600) HOUR,MIN,SEC
 1600 FORMAT("0",5X,2I2,":",F6.3," IS THE TIME TO END PROCESSING.")
C
C  READ CALIBRATION LEVELS AND PRINT OUT.
C
C  USE A SEPARATE HEADING FOR ORDINARY AND EXTRA-ORDINARY PROCESSING.
C
      READ(1,1700) (VOLTS(I),I=1,6)
 1700 FORMAT(6(F7.2/))
      IF(XCHANL.GE.4) GOTO 10
C
C  PRINT CALIBRATION LEVELS AND HEADING FOR EXTRA-ORDINARY
C  PROCESSING.
C
      WRITE(3,1800) (VOLTS(I),I=1,6)
 1800 FORMAT("0",5X,"EXTRA-ORDINARY CALIBRATION LEVELS:"/
     *"0",19X,"-50",2X,F7.2/" ",19X,"-40",2X,F7.2/
     *" ",19X,"-30",2X,F7.2/" ",19X,"-20",2X,F7.2/
```

```
     *" ",19X,"-10",2X,F7.2/" ",19X,"  0",2X,F7.2)
      WRITE(3,1900)
 1900 FORMAT("1",5X,"SECONDS FROM LAUNCH",5X,"AVERAGE",5X,"D.A. RATE"/
     *" ",26X,"DIGITAL VALUE",13X,"EX-ORD LEVEL"/)
      GOTO 20
C
C  PRINT CALIBRATION LEVELS AND HEADING FOR ORDINARY
C  PROCESSING.
C
   10 WRITE(3,2000) (VOLTS(I),I=1,6)
 2000 FORMAT("0",5X,"ORDINARY CALIBRATION LEVELS:"/
     *"0",13X,"-50",2X,F7.2/" ",13X,"-40",2X,F7.2/
     *" ",13X,"-30",2X,F7.2/" ",13X,"-20",2X,F7.2/
     *" ",13X,"-10",2X,F7.2/" ",13X,"  0",2X,F7.2)
      WRITE(3,2100)
 2100 FORMAT("1",5X,"SECONDS FROM LAUNCH",5X,"AVERAGE",5X,"D.A. RATE"/
     *" ",26X,"DIGITAL VALUE",15X,"ORD LEVEL"/)
C
C  CONVERT START AND END TIMES TO SECONDS FROM LAUNCH.
C  LOCATE RECORD IN WHICH PROCESSING IS TO BEGIN.
C
   20 IF(STARTM.GE.ENDTIM) GOTO 110
      ENDTIM=ENDTIM-LAUNCH
      STARTM=STARTM-LAUNCH
      SRCHTM=STARTM-0.5
   30 READ(4) RCDSEC,(DATA(I),I=1,5000)
      IF(EOF(4).NE.0.0) GOTO 80
      IF(SRCHTM.LT.RCDSEC.AND.(.NOT.OPENFLG)) GOTO 90
      OPENFLG=.TRUE.
      IF(SRCHTM.GT.(RCDSEC+1.0)) GOTO 30
C
C  CALCULATE LAST ELEMENT OF THE FRAME IMMEDIATELY PRECEDING
C  THE FRAME WHICH CONTAINS THE ARRAY ELEMENT CORRESPONDING
C  TO THE SEARCH TIME (START TIME MINUS .5 SECOND).
C
   40 CALL FINDPT(SRCHTM,RCDSEC,INDEX)
C
C  FILL ARRAY XPWR WITH ONE SECOND (1000 POINTS) OF DATA.
C  IF A DATA VALUE IS LESS THEN ZERO OR GREATER THAN 4095
C  (12 BIT DIGITIZER RESOLUTION) THEN FLAG THAT POINT AS
C  BAD DATA AND LEAVE OUT OF AVERAGE.
C
      NSMPLS=1
   50 IF(INDEX.GE.5000) GOTO 60
      XINDEX=INDEX+XCHANL
      IF(DATA(XINDEX).LT.0.OR.DATA(XINDEX).GT.4095) CALL BADDATA
      IF(GOODFLG) XPWR(NSMPLS-NBADPTS)=DATA(XINDEX)
      INDEX=INDEX+5
      NSMPLS=NSMPLS+1
      GOODFLG=.TRUE.
      IF(NSMPLS.GT.1000) GOTO 70
      GOTO 50
C
C  REFILL ARRAY DATA WITH ONE SECOND OF DATA.
```

```
C
   60 READ(4) RCDSEC,(DATA(I),I=1,5000)
      IF(EOF(4).NE.0.0) GOTO 100
      IF(RCDSEC.GE.ENDTIM) ENDFLG=.TRUE.
      INDEX=0
      GOTO 50
C
C  COMPUTE AVERAGE VALUE OF DATA.
C
   70 CALL AVERAG((NSMPLS-1-NBADPTS),STARTM,VOLTS)
      IF(ENDFLG) STOP
      NSMPLS=1
      GOTO 50
C
C  COME HERE IF AN END OF FILE IS REACHED ON THE DATA FILE
C  BEFORE THE START TIME IS FOUND.
C
   80 WRITE(3,2200)
 2200 FORMAT("0",5X,"END OF FILE ON TAPE BEFORE START TIME REACHED.")
      STOP1
C
C  COME HERE IF THE START TIME IS LESS THAN THE TIME OF THE
C  FIRST DATA RECORD.
C
   90 WRITE(3,2300)
 2300 FORMAT("0",5X,"START TIME LESS THAN FIRST DATA RECORD TIME ON TAPE
     *.")
      OPENFLG=.TRUE.
      STARTM=RCDSEC+1.0
      SRCHTM=RCDSEC+0.5
      GOTO 40
C
C  COME HERE IF AN END OF FILE WAS REACHED ON THE DATA FILE
C  BEFORE THE STOP TIME WAS FOUND.
C
  100 WRITE(3,2400) RCDSEC
 2400 FORMAT("0",5X,"END OF FILE ON TAPE BEFORE END TIME REACHED."/
     *" ",5X,"TIME OF LAST DATA RECORD READ WAS",F6.1,".")
      STOP2
C
C  COME HERE IF THE START TIME IS GREATER THEN OR EQUAL TO
C  THE STOP TIME.
C
  110 WRITE(3,2500)
 2500 FORMAT("0",5X,"STARTING TIME IS GREATER THAN OR EQUAL TO ENDING TI
     *ME."/"0",5X,"EXECUTION IS TERMINATED.")
      STOP3
      END
      SUBROUTINE AVERAG(ISTOP,TSS,VOLTS)
C
C  THIS SUBROUTINE COMPUTES THE AVERAGE OF ARRAY XPWR
C  WHICH CONTAINS ISTOP ELEMENTS AND CONVERTS THE DIGITAL
C  VALUE TO A DECIBEL VALUE.
C
```

```
      REAL ORD,TDELT,TEST,TSS,VOLTS(6)
      LOGICAL GOODFLG,SUMFLG
      INTEGER ISTOP,NBADPTS,XPWR(1000)
      COMMON/BAD/GOODFLG,NBADPTS
      COMMON/DARATE/ORD,SUMFLG,TDELT,XPWR
C
C  COMPUTE AVERAGE.
C
      TEST=0.0
      DO 10 I=1,ISTOP
         TEST=TEST+FLOAT(XPWR(I))
   10 CONTINUE
      TEST=TEST/ISTOP
C
C  CONVERT AVERAGE DATA VALUE TO DB POWER LEVEL.
C
      CALL CONVRT(TEST,TSS,VOLTS)
      RETURN
      END
      SUBROUTINE CONVRT(TEST,TSS,VOLTS)
C
C  THIS SUBROUTINE CONVERTS DATA VALUE TEST TO A DB POWER
C  LEVEL VOLTS.
C
      REAL ORD,TDELT,TEST,VOLTS(6)
      LOGICAL GOODFLG,SUMFLG
      INTEGER NBADPTS,XPWR(1000)
      COMMON/BAD/GOODFLG,NBADPTS
      COMMON/DARATE/ORD,SUMFLG,TDELT,XPWR
C
C  LOCATE DB LEVELS WHICH BRACKET THE DATA VALUE.
C
      I=1
      IF(TEST.LT.VOLTS(I)) GOTO 30
   10 I=I+1
      IF(TEST.LT.VOLTS(I)) GOTO 20
      IF(I.NE.6) GOTO 10
      EXORD=0.
      GOTO 40
C
C  COMPUTE INTERPOLATED DB LEVEL.
C
   20 X=10.*(TEST-VOLTS(I))/(VOLTS(I-1)-VOLTS(I))
      EXORD=-60.+FLOAT(10*I)-X
      GOTO 40
   30 EXORD=-50.
   40 IF(SUMFLG) GOTO 50
C
C  IF FIRST TIME THROUGH, A POWER DIFFERENCE CANNOT BE
C  COMPUTED.  THEREFORE PRINT OUT *'S.
C
      SUMFLG=.TRUE.
      ORD=EXORD
      WRITE(3,1000) TSS
```

```
 1000 FORMAT(13X,F6.2,25X,"*.****")
      GOTO 60
C
C  COMPUTE DB DIFFERENCE.
C
   50 EXXOR=EXORD
      EXORD=EXXOR-ORD
C
C  PRINT OUT TIME AND DB DIFFERENCE.
C
      WRITE(3,1100) TSS,EXORD
 1100 FORMAT(13X,F6.2,23X,F7.3)
      ORD=EXXOR
   60 WRITE(3,1200) TEST,ORD
 1200 FORMAT(29X,F8.1,18X,F8.3/)
C
C  INCREMENT TIME FOR NEXT PASS.
C
      TSS=TSS+TDELT
C
C  PRINT OUT NUMBER OF BAD DATA POINTS WHICH OCCURRED DURING
C  THE LAST SECOND.
C
      IF(NBADPTS.EQ.0) RETURN
      WRITE(3,1300) NBADPTS
 1300 FORMAT("+",55X,"THERE WERE",I3," BAD DATA POINTS DURING THE LAST S
     *ECOND.")
      NBADPTS=0
      RETURN
      END
      SUBROUTINE FINDPT(SRCHTM,RCDTIM,POINT)
C
C  THIS SUBROUTINE LOCATES THE LAST ARRAY ELEMENT OF THE FRAME
C  WHICH PRECEDES THE FRAME THAT CONTAINS THE ARRAY ELEMENT
C  CORRESPONDING TO THE SEARCH TIME.
C
      REAL RCDTIM,SRCHTM
      INTEGER POINT
      POINT=INT((SRCHTM-RCDTIM)*5000.0)
      POINT=POINT-MOD(POINT,5)
      RETURN
      END
      SUBROUTINE READTM(TIME,UNIT)
C
C  THIS SUBROUTINE READS IN TIME VALUES ENCODED IN HOURS,
C  MINUTES AND SECONDS AND CONVERTS THIS TIME TO SECONDS.
C
      REAL SEC,TIME
      INTEGER HOUR,MIN,UNIT
      COMMON/TIME/HOUR,MIN,SEC
      READ(UNIT,1000) HOUR,MIN,SEC
 1000 FORMAT(2I2,1X,F6.3)
      TIME=3600.0*FLOAT(HOUR)+60.0*FLOAT(MIN)+SEC
      RETURN
```

```
      END
      SUBROUTINE BADDATA
C
C  THIS SUBROUTINE KEEPS TRACK OF THE NUMBER OF BAD DATA VALUES
C  (<0 OR >4095).
C ----------------------------
      INTEGER NBADPTS
      LOGICAL GOODFLG
      COMMON/BAD/GOODFLG,NBADPTS
      NBADPTS=NBADPTS+1
      GOODFLG=.FALSE.
      RETURN
      END
```

APPENDIX IX. Listing of CDC Fortran program DA2NE.

The procedure file used to run this program as a batch job on the Cyber is nearly identical to the one used to run program FR2NE (Appendix V). FR2NE has been replaced by DA2NE and the experimental Faraday rotation rates (the seventh parameter on a data line) have been replaced by the experimental differential absorption rates.

Subroutines COEFF, FIELD, SENWYL, PRESSR, MONTH, INTER and ROUND are the same as those used in program FR2NE and are not included in this appendix. All subroutines which are not pre-compiled and placed in user library BLIB must be included when compiling DA2NE.

```
/JOB
/NOSEQ
MKM.
SIGNON(3MIKEKM)
BILL,ELEC,PS2714.
USE,OPTION,DA2NE,BLIB.
GET,OPTION.
PRINT.
GET,DA2NE,BLIB.
ADDLIB,BLIB.
FTN,I=DA2NE,L=0,A.
LGO,,BODY,HEADER.
REWIND,HEADER,BODY.
SKIPR,HEADER.
COPYEI,BODY,HEADER.
PACK,HEADER.
PRINT/NORIGHT,HEADER.
/EOR
18.1020   1979.15068  5040000.   630000.
   32.2826    73.1569     49458. 1285.0000    50.9989  -93.4357    -0.0001
   31.7055    73.2503     50696. 1273.0000    51.0018  -93.4354    -0.0001
   31.1363    73.3381     51923. 1262.0000    51.0046  -93.4351    -0.0250
   30.6291    73.4249     53140. 1247.0000    51.0074  -93.4348    -0.1250
   30.2047    73.5037     54345. 1238.0000    51.0101  -93.4343    -0.1880
   29.8226    73.5658     55541. 1231.0000    51.0128  -93.4335    -0.1120
   29.4180    73.6137     56727. 1224.0000    51.0156  -93.4329     0.0380
   28.9623    73.6549     57901. 1213.0000    51.0186  -93.4325     0.1750
   28.4880    73.6930     59065. 1203.0000    51.0215  -93.4322     0.2500
   28.0555    73.7311     60213. 1185.0000    51.0244  -93.4319     0.3380
   27.6927    73.7730     61357. 1182.0000    51.0271  -93.4314     0.4750
   27.3744    73.8130     62505. 1190.0000    51.0298  -93.4308     0.7250
   27.0603    73.8427     63652. 1185.0000    51.0325  -93.4302     1.1750
   26.7380    73.8658     64783. 1160.0000    51.0353  -93.4296     1.2380
   26.4214    73.8904     65893. 1147.0000    51.0380  -93.4291     1.9000
```

```
      PROGRAM DA2NE(INPUT,OUTPUT,HEADNG,TAPE1=INPUT,TAPE2=OUTPUT,
     *              TAPE3=HEADNG)
```

```
C
C  ANALYSIS OF DIFFERENTIAL ABSORPTION WITHOUT FARADAY ROTATION
C  GIVEN A MODEL FOR THE COLLISION FREQUENCY.
C
C  THE FIRST INPUT CARD CONTAINS FLIGHT INFORMATION.
C
C          COLUMNS  1-10        ROCKET NUMBER
C          COLUMNS 11-20        DATE (IN DECIMAL PARTS OF YEARS
C                                     E.G. FEBRUARY 26,1979 = 1979.15616)
C          COLUMNS 21-30        FREQUENCY (HERTZ)
C          COLUMNS 31-40        COLLISION FREQUENCY MODEL
C                               PARAMETER (HERTZ/PASCAL)
C          FORMAT(F10.4,F10.5,F10.0,F10.0)
C
C  THE REMAINING DATA CARDS CONTAIN THE ROCKET LOCATION AND
C  VELOCITY ALONG WITH THE EXPERIMENTAL DIFFERENTIAL ABSORPTION RATE.
C
C          COLUMNS  1-10        AZIMUTH (DEGREES)
C          COLUMNS 11-20        ELEVATION (DEGREES)
C          COLUMNS 21-30        HEIGHT (METERS)
C          COLUMNS 31-40        TOTAL VELOCITY (METERS/SECOND)
C          COLUMNS 41-50        ROCKET LATITUDE (DEGREES)
C          COLUMNS 51-60        ROCKET LONGITUDE (DEGREES)
C          COLUMNS 61-70        DIFFERENTIAL ABSORPTION
C                               (EXPERIMENTAL VALUE)(DECIBELS/SECOND)
C          FORMAT(2F10.4,F10.0,4F10.4)
C
C                        **** IMPORTANT NOTE ****
C  THE PRESSURE IS COMPUTED IN A USER LIBRARY SUBROUTINE NAMED
C  PRESSR.  MAKE SURE THAT YOU HAVE EITHER CALLED THE LIBRARY OR
C  HAVE MADE IT A LOCAL SUBROUTINE BY INSERTING IT AT THE
C  END OF THIS PROGRAM.
C
C  MODIFIED BY M K MCINERNEY OCTOBER 1979.
C
C  COMMON VARIABLES TO BE USED IN DIFFERENTIAL ABSORPTION RATE
C  SUBPROGRAM, MDA, AND COMMON VARIABLES TO BE USED IN SUBROUTINE
C  NPRNT, WHICH PRINTS OUT THE VARIABLE VALUES.
C
      REAL AC,CF,ED,ED1,ED2,FC,MDA,S,W,X,Z
      COMMON/AREA1/S,AC
      COMMON/AREA2/ED,X,Z,AZD,BN,F,HT,ELD,BD,RLATD,BW,RLNGD,B,DAE,
     *V,Y,TH,DA,FC,FR,FRE
      COMMON/AREA3/W,CF,ROR,RIO,RXR,RIX,ROI,AIO,RXI,AIX,CTH
      COMMON/COEFFS/GAUSS(11,11)
C
C  ENTER PARAMETERS OF OBLATE EARTH.
C
      A=6.378165E6
      A2=A*A
      A4=A2*A2
      FLAT=1.-1./298.3
      B2=(A*FLAT)**2
      A2B2=A2*(1.-FLAT**2)
```

```
      A4B4=A4*(1.-FLAT**4)
C
C  ENTER FLIGHT PARAMETERS.
C
      READ(1,1000) RKTNO,TM,F,CFM
 1000 FORMAT(F10.4,F10.5,F10.0,F10.0)
      WRITE(3,1100)
 1100 FORMAT("1INPUT DATA DECK:")
      WRITE(3,1200) RKTNO,TM,F,CFM
 1200 FORMAT("-ROCKET NUMBER     DATE     FREQUENCY     COLLISION FREQUE
     *NCY MODEL PARAMETER"/"0    ",F7.4,5X,F10.5,1X,F10.0,17X,F10.0/"0")
      WRITE(3,1300)
 1300 FORMAT(3X,"AZIMUTH   ELEVATION   ALTITUDE  VELOCITY  LATITUDE   LO
     *NGITUDE        EXPERIMENTAL DIFFERENTIAL ABSORPTION RATE        PR
     *ESSURE"/"0")
C
C  INITIALIZE THE COEFFICIENTS USED IN CALCULATING THE
C  MAGNETIC FIELD, COMPUTE THE RADIAN FREQUENCY AND
C  DEFINE A DEGREES-TO-RADIANS CONVERSION VARIABLE.
C
      CALL COEFF(TM)
      W=6.2831853*F
      DTR=1.745329E-2
C
C  ENTER VARIABLES.
C
   10 READ(1,1400) AZD,ELD,HT,V,RLATD,RLNGD,DAE
 1400 FORMAT(2F10.4,F10.0,4F10.4)
      IF(EOF(1).NE.0.0) GOTO 120
      CALL PRESSR (RLATD,HT,TM,P)
      WRITE(3,1500) AZD,ELD,HT,V,RLATD,RLNGD,DAE,P
 1500 FORMAT(2X,F9.4,2X,F9.4,4X,F7.0,4X,F5.0,3X,F8.4,3X,F9.4,22X,F10.4,
     *23X,F10.5/" ")
      FRE=0.0
      CF=CFM*P
C
C  CONVERT DEGREES TO RADIANS.
C
      AZ=AZD*DTR
      EL=ELD*DTR
      RLATR=RLATD*DTR
      RLNGR=RLNGD*DTR
C
C  FIND GEOCENTRIC COORDINATES OF ROCKET.
C
      SINLA=SIN(RLATR)
      SINLA2=SINLA*SINLA
      COSLA2=1.-SINLA2
      DEN2=A2-A2B2*SINLA2
      DEN=SQRT(DEN2)
      FAC=(((HT*DEN)+A2)/((HT*DEN)+B2))**2
      CT=SINLA/SQRT(FAC*COSLA2+SINLA2)
      R=SQRT(HT*(HT+2.*DEN)+(A4-A4B4*SINLA2)/DEN2)
      ST=SQRT(1.-CT**2)
```

```
C
C  CALCULATE GEOMAGNETIC FIELD AT ROCKET.
C
      SPH=SIN(RLNGR)
      CPH=COS(RLNGR)
      CALL FIELD (R,ST,CT,SPH,CPH,BR,BT,BP,B)
C
C  TRANSFORM FIELD COMPONENTS, GEOCENTRIC TO GEODETIC.
C
      SIND=SINLA*ST-SQRT(COSLA2)*CT
      COSD=SQRT(1.-SIND**2)
      BN=-BT*COSD-BR*SIND
      BD=BT*SIND-BR*COSD
      BW=-BP
      S=-1.758803E11*B
      Y=S/W
C
C  CALCULATE COSINE OF PROPAGATION ANGLE AND PROPAGATION ANGLE.
C
      CEL=COS(EL)
      CTH=(CEL*COS(AZ)*BN-SIN(EL)*BD-CEL*SIN(AZ)*BW)/B
      TH=ACOS(CTH)/DTR
C
C  CALCULATE FR AND DA COEFFICIENTS.
C
      FV=F*V
      FC=6.004153E-7*FV
      AC=1.820428E-7*FV
C
C  ITERATE TO MATCH DAE.
C
      WRITE(2,1600) RKTNO,TM,CFM
 1600 FORMAT("1  ",F10.4,F10.3,1PE10.2)
C
C  FOR AN INITIAL GUESS AS TO THE ELECTRON DENSITY WHICH CORRESPONDS
C  TO THE EXPERIMENTAL DIFFERENTIAL ABSORPTION RATE, LOCATE IN
C  WHICH ELECTRON DENSITY DECADE BETWEEN 1.0E00 AND 1.0E20 THE
C  EXPERIMENTAL DIFFERENTIAL ABSORPTION RATE FALLS.
C
C  IF THE THEORETICAL DIFFERENTIAL ABSORPTION RATE IS DISCONTINUOUS
C  WITHIN THE DECADE SEPARATE THE ANALYSIS (CHECK) BASED UPON THE
C  TYPE OF DISCONTINUITY.
C
C     FOR A NEGATIVE-GOING DISCONTINUITY, CHECK TO SEE IF THE EXP DA
C     RATE IS GREATER THAN (OR EQUAL TO) THE MINIMUM NEGATIVE
C     THEORETICAL DA RATE.
C     FOR A POSITIVE-GOING DISCONTINUITY, CHECK TO SEE IF THE EXP DA
C     RATE IS LESS THAN (OR EQUAL TO) THE MAXIMUM POSITIVE
C     THEORETICAL DA RATE.
C
      ED1=1.0E+00
      DA1=MDA(ED1)
   20 ED2=10.0*ED1
      IF(ED2.GT.1.0E+20) GOTO 110
```

```
      DA2=MDA(ED2)
      IF((DA1*DA2).LE.0.0) GOTO 30
      IF(((DAE-DA2)*(DAE-DA1)).LE.0.0) GOTO 50
      GOTO 40
   30 IF(DA1.GT.DA2.AND.DAE.GE.DA2) GOTO 50
      IF(DA1.LT.DA2.AND.DAE.LE.DA2) GOTO 50
   40 ED1=ED2
      DA1=DA2
      GOTO 20
C
C  ITERATE BY HALVING INTERVAL TO MATCH THE EXPERIMENTAL DIFFERENTIAL
C  ABSORPTION RATE.
C
C  IF THE THEORETICAL DA INTERVAL IS DISCONTINUOUS THEN SEPARATE THE
C  ANALYSIS INTO TWO CASES.  ANALYZE BASED UPON WHICH THEORETICAL DA
C  RATE END VALUE HAS THE SAME SIGN AS THE THEORETICAL DA RATE WHICH
C  CORRESPONDS TO THE CENTER ED VALUE.
C
   50 DO 100 I=1,40
         ED=(ED1+ED2)/2.0
         DA=MDA(ED)
         IF(MOD(I,2).EQ.0) CALL NPRNT
         IF((DA1*DA2).LE.0.0) GOTO 70
   60    IF(((DAE-DA)*(DAE-DA1)).LE.0.0) GOTO 90
         GOTO 80
   70    IF((DA1*DA).GT.0.0) GOTO 60
         IF(((DAE-DA)*(DAE-DA2)).GT.0.0) GOTO 90
   80    ED1=ED
         DA1=DA
         GOTO 100
   90    ED2=ED
         DA2=DA
  100 CONTINUE
      GOTO 10
C
C  PRINT OUT ERROR DIAGNOSTIC IF EXPERIMENTAL DIFFERENTIAL ABSORPTION
C  VALUE WAS UNCOMPARABLE.
C
  110 WRITE(2,1700) DAE
 1700 FORMAT("OELECTRON DENSITIES BETWEEN 1.0E00 AND 1.0E20 HAVE BEEN SC
     *ANNED."/"ONO DECADE WAS FOUND WHICH CONTAINED A DIFFERENTIAL ABSOR
     *PTION VALUE"/"OOF ",F10.5,"."/"OTHIS DIFFERENTIAL ABSORPTION VALUE
     * IS PROBABLY UNMATCHABLE.")
      GOTO 10
  120 WRITE(3,1300)
      WRITE(3,1800)
      WRITE(3,1900)
 1800 FORMAT("1NOTE THE FOLLOWING ABREVIATIONS AND CONVENTIONS:"/
     *"_  AZ  - AZIMUTH OF ROCKET FROM TRANSMITTER  - - - - DEGREES"/
     *"   EL  - ELEVATION OF ROCKET FROM TRANSMITTER  - - - DEGREES"/
     *"   LAT - GEODETIC LATITUDE OF ROCKET  - - - - - - - DEGREES"/
     *"   LNG - ROCKET LONGITUDE - - - - - - - - - - - - - DEGREES"/
     *"   V   - TOTAL VELOCITY - - - - - - - - - - - METERS/SECOND"/
     *"   BN  - GEOMAGNETIC FLUX (NORTH COMPONENT) - - - -   TESLA"/
```

```
     *"   BD  - GEOMAGNETIC FLUX (DOWNWARD COMPONENT)  - - -   TESLA"/
     *"   BW  - GEOMAGNETIC FLUX (WESTWARD COMPONENT)  - - -   TESLA"/
     *"   B   - TOTAL GEOMAGNETIC FLUX - - - - - - - - -   TESLA"/
     *"   Y   - RATIO OF GYRO TO WAVE FREQUENCY - - - - - - - - "/
     *"   ROR - POLARIZATION (ORDINARY)(REAL PART) - - - - - - - "/
     *"   RXR - POLARIZARION (EXTRAORDINARY)(REAL PART)  - - - - "/
     *"   ROI - POLARIZATION (ORDINARY)(IMAGINARY PART)  - - - - "/
     *"   RXI - POLARIZATION (EXTRAORDINARY)(IMAGINARY PART) - - - "/
     *"   CTH - COSINE OF TH  - - - - - - - - - - - - - - - - - - ")
 1900 FORMAT(
     *"   RIO - REFRACTIVE INDEX (ORDINARY)  - - - - - - - - - - "/
     *"   RIX - REFRACTIVE INDEX (EXTRAORDINARY) - - - - - - - - "/
     *"   AIO - ABSORPTION INDEX (ORDINARY)  - - - - - - - - - - "/
     *"   AIX - ABSORPTION INDEX (EXTRAORDINARY) - - - - - - - - "/
     *"   TH  - ANGLE BETWEEN WAVE NORMAL & GEOMAGNETIC FIELD  DEGREES"/
     *"   F   - FREQUENCY  - - - - - - - - - - - - - - -   HERTZ"/
     *"   FRE - FARADAY RATE, EXPERIMENTAL - - - -   DEGREES/SECOND"/
     *"   FR  - FARADAY RATE, COMPUTATIONAL   - - -   DEGREES/SECOND"/
     *"   DAE - DIFFERENTIAL ABSORBTION RATE, EXPERIMENTAL  DBS/SECOND"/
     *"   DA  - DIFFERENTIAL ABSORPTION RATE, COMPUTATIONAL DBS/SECOND"/
     *"   HT  - ALTITUDE - - - - - - - - - - - - - - - - -  METERS"/
     *"   ED  - ELECTRON DENSITY  - - - - - - - - - - METERS**-3"/
     *"   X   - RATIO OF PLASMA TO WAVE FREQUENCY  - - - - - - - "/
     *"   CF  - COLLISION FREQUENCY - - - - - - - - - - -   HERTZ"/
     *"   Z   - RATIO OF COLLISION FREQUENCY TO ANGULAR WAVE FREQUENCY")
      STOP
      END
      REAL FUNCTION MDA(ED)
C
C  THIS FUNCTION COMPUTES THE DIFFERENTIAL ABSORPTION RATE AS A
C  FUNCTION OF ELECTRON DENSITY.
C
      REAL AC,AIO,AIX,CF,CTH,ED,RIO,RIX,ROI,ROR,RXI,RXR,S,W
      COMMON/AREA1/S,AC
      COMMON/AREA3/W,CF,ROR,RIO,RXR,RIX,ROI,AIO,RXI,AIX,CTH
      CALL SENWYL(ED,CF,S,CTH,W,RIO,RIX,AIO,AIX,ROR,RXR,ROI,RXI)
      MDA=AC*(AIX-AIO)
      RETURN
      END
      SUBROUTINE NPRNT
C
C  THIS SUBROUTINE PRINTS OUT THE VALUES OF THE VARIABLES USED IN
C  FINDING THE ELECTRON DENSITY.
C
      REAL AIO,AIX,AZD,B,BD,BN,BW,CF,CTH,DA,DAE,ED,ELD,F,FC,FR,FRE,
     *HT,RIO,RIX,RLATD,RLNGD,ROI,ROR,RXI,RXR,TH,V,W,X,Y,Z
      COMMON/AREA2/ED,X,Z,AZD,BN,F,HT,ELD,BD,RLATD,BW,RLNGD,B,DAE,
     *V,Y,TH,DA,FC,FR,FRE
      COMMON/AREA3/W,CF,ROR,RIO,RXR,RIX,ROI,AIO,RXI,AIX,CTH
C
C  CALCULATE ROTATION AND REFRACTION INDICES.
C
      FR=FC*(RIO-RIX)
      X=3182.6018*ED/W/W
```

```
      Z=CF/W
      WRITE(2,1000)
      WRITE(2,1100) AZD,BN,ROR,RIO,F,HT
      WRITE(2,1200) ELD,BD,RXR,RIX,FRE,ED
      WRITE(2,1300) RLATD,BW,ROI,AIO,FR,X
      WRITE(2,1400) RLNGD,B,RXI,AIX,DAE,CF
      WRITE(2,1500) V,Y,CTH,TH,DA,Z
1000 FORMAT(" ")
1100 FORMAT("     AZ ",1PE13.6,"     BN ",1PE13.6,"     ROR ",1PE13.6,
    *      "    RIO ",1PE13.6,"      F ",1PE13.6,"      HT ",1PE13.6)
1200 FORMAT("     EL ",1PE13.6,"     BD ",1PE13.6,"     RXR ",1PE13.6,
    *      "    RIX ",1PE13.6,"    FRE ",1PE13.6,"      ED ",1PE20.13)
1300 FORMAT("    LAT ",1PE13.6,"     BW ",1PE13.6,"     ROI ",1PE13.6,
    *      "    AIO ",1PE13.6,"     FR ",1PE13.6,"       X ",1PE13.6)
1400 FORMAT("    LNG ",1PE13.6,"      B ",1PE13.6,"     RXI ",1PE13.6,
    *      "    AIX ",1PE13.6,"    DAE ",1PE13.6,"      CF ",1PE13.6)
1500 FORMAT("      V ",1PE13.6,"      Y ",1PE13.6,"     COS ",1PE13.6,
    *      "     TH ",1PE13.6,"     DA ",1PE13.6,"       Z ",1PE13.6)
      RETURN
      END
```

APPENDIX X. Listing of CDC Fortran IV program PC2ED.

This program was run as a batch job on the Cyber using the procedure file which makes up the first seventeen lines of this appendix. Following the /EOR statement of this procedure file are the three input parameters: the first N/I scale factor; the last time to use the first N/I scale factor; and the second N/I scale factor.

The data file to be converted, PC20S, is the data file which was created by program SWEEP (Appendix II). Because program PC2ED uses the same data file for input and output, the input file, PC20S, is renamed ED20S on line ten so that the output file can be saved without destroying the input file.

```
/JOB
/NOSEQ
MKM.
SIGNON(3MIKEKM)
BILL,ELEC,PS2714.
USE,OPTION,PC2ED,PC20S.
GET,OPTION.
PRINT.
GET,PC2ED,PC20S.
RENAME,ED20S=PC20S.
FTN,I=PC2ED,L=0,A.
LGO,,,ED20S.
SAVE,ED20S.
/EOR
 1.6E+16
 291.6
 2.0E+16
```

```
      PROGRAM PC2ED(INPUT,OUTPUT,DATA,TAPE1=INPUT,TAPE2=OUTPUT,
     *              TAPE3=DATA)
C
C  THIS PROGRAM CONVERTS PROBE CURRENT TO ELECTRON DENSITY.
C  THE THREE NUMBERS TO BE ENTERED ARE:
C
C      SCALE - - - THE FIRST N/I VALUE.
C      T - - - - - THE TIME UP TO WHICH SCALE IS TO BE USED.
C                  AFTER T LSCALE WILL BE USED AS THE N/I VALUE.
C      LSCALE - -  THE LAST N/I VALUE.
C
C  THESE PARAMETERS ARE INPUT WITHOUT FORMAT, SO ALL THAT HAS
C  TO BE DONE IS TO PLACE THEM ON DIFFERENT LINES.
C
C                    *****     CAUTION     *****
C
C      THE INPUT DATA FILE IS USED AS THE OUTPUT DATA FILE.
C      THEREFORE INFORMATION IN THE INPUT DATA FILE WILL BE
C      OVERWRITTEN (DESTROYED).
C
```

```
C  WRITTEN BY M K MCINERNEY    NOVEMBER, 1979.
C
      REAL PC(6000),TIM(1200),LSCALE,SCALE
      INTEGER MAXI,MAXJ,M,N
C
C  READ IN PROBE CURRENT VERSUS TIME VALUES.
C
      CALL PREAD(TIM,PC,3,MAXI,MAXJ)
C
C  READ IN CONVERSION FACTORS.
C
      READ*, SCALE
      READ*, T
      READ*, LSCALE
C
C  DETERMINE WHICH PROBE CURRENT VALUE CORRESPONDS TO TIME T.
C
      M=INT((((T-TIM(1))/(TIM(2)-TIM(1)))*5.0)+0.5)+1
      IF(M.LT.1) M=1
      IF(M.GT.MAXJ) M=MAXJ
C
C  CONVERT FIRST PORTION OF PROBE CURRENT ARRAY.
C
      DO 10 I=1,M
         IF(PC(I).EQ.1.0.OR.PC(I).EQ.2.0) GOTO 10
         PC(I)=SCALE*PC(I)
   10 CONTINUE
C
C  CONVERT SECOND PORTION OF PROBE CURRENT ARRAY.
C
      IF(M.LT.MAXJ) M=M+1
      DO 20 I=M,MAXJ
         IF(PC(I).EQ.1.0.OR.PC(I).EQ.2.0) GOTO 20
         PC(I)=LSCALE*PC(I)
   20 CONTINUE
C
C  OUTPUT CONVERTED ARRAY TO THE SAME FILE AS USED FOR INPUT.
C
      CALL PWRITE(TIM,PC,3,MAXI,MAXJ)
      STOP
      END
      SUBROUTINE PREAD(TIM,PC,NN,MAXI,MAXJ)
C
C  THIS SUBROUTINE READS IN DATA WITH A FORMAT OF
C  ONE TIME VALUE (F7.1) FOLLOWED BY FIVE ELECTRON
C  DENSITY VALUES IN EXPONENTIAL FORM (E13.4).
C
C     TIM IS THE ARRAY OF TIME VALUES.
C     ED IS THE ARRAY OF ELECTON DENSITY VALUES.
C     MAXJ IS THE NUMBER OF ELECTRON DENSITY VALUES.
C     MAXI IS THE NUMBER OF TIME VALUES.
C     NN IS THE UNIT (TAPE) NUMBER TO BE READ.
C
      REAL TIM(1200),PC(6000)
```

```
      INTEGER NN,MAXI,MAXJ
      REWIND NN
      I=0
      J=0
   10 I=I+1
      J=J+5
      JL=J-4
      READ(NN,1000) TIM(I),(PC(K),K=JL,J)
      IF(EOF(NN)) 20,10
 1000 FORMAT(F7.1,5E13.4)
   20 MAXI=I-1
      MAXJ=J-5
      RETURN
      END
      SUBROUTINE PWRITE(TIM,PC,NN,MAXI,MAXJ)
C
C  THIS SUBROUTINE WRITES OUT THE TIME AND ELECTON DENSITY
C  ARRAYS IN THE FORMAT USED IN SUBROUTINE PREAD.
C
      REAL TIM(1200),PC(6000)
      INTEGER NN,MAXI,MAXJ
      REWIND NN
      I=0
      J=0
   10 I=I+1
      IF(I.GT.MAXI) RETURN
      J=J+5
      JL=J-4
      WRITE(NN,1000) TIM(I),(PC(K),K=JL,J)
 1000 FORMAT(F7.1,5E13.4)
      GOTO 10
      RETURN
      END
```

APPENDIX XI. Listing of CDC Fortran IV program EDINTER.

The first thirteen lines of this appendix are the Cyber statements which were used to run the program as a batch job.

The data file which has gaps to be filled, ED20S, is renamed on line ten to avoid destroying it after program execution.

```
/JOB
/NOSEQ
MKM.
SIGNON(3MIKEKM)
BILL,ELEC,PS2714.
USE,OPTION,EDINTER,ED20S.
GET,OPTION.
PRINT.
GET,EDINTER,ED20S.
RENAME,ED20=ED20S.
FTN,I=EDINTER,L=0,A.
LGO,,,ED20.
SAVE,ED20.
```

```
      PROGRAM EDINTER(INPUT,OUTPUT,DATA,TAPE1=INPUT,TAPE2=OUTPUT,
     *                 TAPE3=DATA)
C
C  THIS PROGRAM IS USED TO FILL IN THE ELECTRON DENSITY
C  PROFILE WHERE THE SWEEPS WERE REMOVED.  THE PROGRAM
C  SEARCHES FOR VALUES OF 2.0 IN THE ELECTRON DENSITY
C  DATA AND WHEN IT FINDS THEM, REPLACES THEM BY LINEARLY
C  INTERPOLATED ELECTRON DENSITY VALUES DERIVED FROM THE
C  TWO ELECTRON DENSITY VALUES BRACKETING THE TWO'S.
C
C  NOTE THAT IF THE FIRST OR LAST N VALUES OF ELECTRON
C  DENSITY ARE TWO'S, THEY CAN NOT BE REPLACED WITH
C  INTERPOLATED VALUES.
C
C                       *****     CAUTION     *****
C
C      THE INPUT DATA FILE IS USED AS THE OUTPUT DATA FILE.
C      THEREFORE INFORMATION IN THE INPUT DATA FILE WILL BE
C      OVERWRITTEN (DESTROYED).
C
C  WRITTEN BY M K MCINERNEY    NOVEMBER, 1979.
C
      REAL TIM(1200),ED(6000),DIFFER
      INTEGER NTIMS,NEDS,FTWO,LTWO,NED
      COMMON/ALL/TIM,ED,NEDS,NTIMS
      DATA TIM/1200*0.0/
      DATA ED/6000*1.0/
      NED=0
C
C  INPUT ELECTRON DENSITY VERSUS TIME VALUES.
```

```
C
      CALL PREAD(3)
      IF(NTIMS.NE.0) GOTO 10
      PRINT*, "*  THERE IS NO DATA IN THE INPUT FILE."
      PRINT*, "   "
      PRINT*, "*  EXECUTION IS TERMINATED."
      STOP 1
C
C  PRINT OUT WARNING IF THERE ARE MORE THAN 6000
C  ELECTRON DENSITY VALUES.
C
   10 IF(NEDS.LE.6000) GOTO 20
      PRINT*, "*  THE GREATEST NUMBER OF ELECTRON DENSITY"
      PRINT*, "   VALUES THIS PROGRAM CAN HANDLE IS 6000."
      PRINT*, " "
      PRINT*, "*  EXECUTION WILL CONTINUE BUT ONLY THE FIRST"
      PRINT*, "   6000 VALUES WILL BE PROCESSED."
C
C  FIND GAPS.
C
   20 CALL FINDTWO(FTWO,LTWO,NED)
      IF(NED.GT.NEDS) GOTO 40
C
C  DO NOT FILL GAP IF A VALUE OF ONE IS ON EITHER SIDE.
C
      IF(ED(FTWO-1).EQ.1.0.OR.ED(LTWO-1).EQ.1.0) GOTO 20
C
C  CALCULATE THE ELECTRON DENSITY INCREMENT TO BE
C  USED OVER THE RANGE OF THE GAP.
C
      DIFFER=(ED(LTWO+1)-ED(FTWO-1))/FLOAT(LTWO-FTWO+2)
C
C  REPLACE GAP WITH INTERPOLATED VALUES.
C
      DO 30 I=FTWO,LTWO
         ED(I)=ED(I-1)+DIFFER
   30 CONTINUE
      GOTO 20
C
C  OUTPUT ELECTRON DENSITY VERSUS TIME VALUES.
C
   40 CALL PWRITE(3)
      STOP
      END
      SUBROUTINE PREAD(NUNIT)
C
C  THIS SUBROUTINE READS IN DATA WITH A FORMAT OF
C  ONE TIME VALUE (F7.1) FOLLOWED BY FIVE ELECTRON
C  DENSITY VALUES IN EXPONENTIAL FORM (E13.4).
C
C     TIM IS THE ARRAY OF TIME VALUES.
C     ED IS THE ARRAY OF ELECTON DENSITY VALUES.
C     NEDS IS THE NUMBER OF ELECTRON DENSITY VALUES.
C     NTIMS IS THE NUMBER OF TIME VALUES.
```

```
C     NUNIT IS THE UNIT (TAPE) NUMBER TO BE READ.
C
      REAL TIM(1200),ED(6000)
      INTEGER NUNIT,EOF,NEDS,NTIMS
      COMMON/ALL/TIM,ED,NEDS,NTIMS
      REWIND NUNIT
      I=0
      J=0
   10 I=I+1
      J=J+5
      JL=J-4
      READ(NUNIT,1000) TIM(I),(ED(K),K=JL,J)
      IF(EOF(NUNIT)) 20,10
 1000 FORMAT(F7.1,5E13.4)
   20 NTIMS=I-1
      NEDS=J-5
      RETURN
      END
      SUBROUTINE FINDTWO(FTWO,LTWO,NED)
C
C  THIS SUBROUTINE LOCATES THE VALUES OF 2.0 IN THE ELECTRON
C  DENSITY ARRAY AND RETURNS THE LOCATION OF THE FIRST AND
C  LAST 2.0 IN EACH SERIES OF TWO'S.
C
C     FTWO IS THE INDEX OF THE FIRST TWO.
C     LTWO IS THE INDEX OF THE LAST TWO.
C     NED IS THE CURRENT INDEX OF THE POINTER IN ARRAY ED.
C
      REAL TIM(1200),ED(6000)
      INTEGER FTWO,LTWO,NED,NEDS,NTIMS
      COMMON/ALL/TIM,ED,NEDS,NTIMS
   10 NED=NED+1
      IF(NED.GT.NEDS) RETURN
      IF(ED(NED).EQ.2.0) GOTO 20
      GOTO 10
   20 FTWO=NED
   30 NED=NED+1
      IF(NED.GT.NEDS) RETURN
      IF(ED(NED).NE.2.0) GOTO 40
      GOTO 30
   40 LTWO=NED-1
      RETURN
      END
      SUBROUTINE PWRITE(NUNIT)
C
C  THIS SUBROUTINE WRITES OUT THE TIME AND ELECTON DENSITY
C  ARRAYS UNDER THE FORMAT USED IN SUBROUTINE PREAD.
C
      REAL TIM(1200),ED(6000)
      INTEGER NUNIT,NEDS,NTIMS
      COMMON/ALL/TIM,ED,NEDS,NTIMS
      REWIND NUNIT
      DO 10 I=1,NTIMS
         J1=5*I-4
```

```
      J2=5*I
      WRITE(NUNIT,1000) TIM(I),(ED(N),N=J1,J2)
1000  FORMAT(F7.1,5E13.4)
  10 CONTINUE
     RETURN
     END
```

APPENDIX XII. Listing of CDC Fortran IV program EDPLOT.

The first twenty-six lines of this appendix list four examples of executing program EDPLOT. File BEDPLT contains the binary execution code for program EDPLOT, which was previously compiled.

The plotting device used in each example is the Zeta drum plotter. This produces a hard copy plot in ink on unlined paper. By changing the GRAB statement on line two of each example plots may be done on a terminal. (e.g. Change ZETA to TEKT to plot on a Tektronix 4006, 4010 or 4112 terminal, or to TK14 to plot on a 4014 terminal.)

The binary plot file, TAPE99, is saved (made permanent) after each execution so that it may be plotted again without re-executing program EDPLOT. This is a precaution in the event of a lost or messed up plot. If permanent file TAPE99 already exists then a REPLACE command must be used, which will destroy the present contents of permanent file TAPE99.

The APPEND command, however, will write a /EOS separator at the end of permanent file TAPE99 and then place the new TAPE99 file after the separator. With multiple appends many plot files can be saved.

A multiple plot file can be separated again by using the COPYBR command. In case the subjects of the plots are forgotten, each individual plot file may be viewed at a Tektronix terminal using the ZETAVU utility.

The first execution sequence listed below was used to plot data file PC20S. Because of the order of files on the BEDPLT execution statement, line four, program EDPLOT interpreted PC20S as DATA1.

The second execution sequence is similar to the first except that data file ED20 was plotted.

Data file ED20 was also plotted in the third execution sequence, only this time a final electron density plot was made. Therefore the PLOT statement, line five, contains the parameters used for an off-line, wide paper, India ink plot.

The fourth execution sequence can be used to plot two data files at once. The BEDPLT statement causes program EDPLOT to interpret file ED20 as DATA1 and file ED21 as DATA2. Because this is a Zeta plot data from both files can be superimposed.

The plot section of program SWEEP (Appendix II) has nearly the same graphics capabilities as program EDPLOT, except that program SWEEP can only plot probe current data files.

```
USE,BEDPLT,PC20S,TRJ20 <CR>
GRAB,GCS(ZETA,3D) <CR>
GET,BEDPLT,PC20S,TRJ20 <CR>
BEDPLT,,,PC20S,,TRJ20 <CR>
PLOT,TAPE99/BIN=49/J=MKMPC20S/LENGTH=60/TIME=30 <CR>
SAVE,TAPE99 <CR>

USE,BEDPLT,ED20,TRJ20 <CR>
GRAB,GCS(ZETA,3D) <CR>
GET,BEDPLT,ED20,TRJ20 <CR>
BEDPLT,,,ED20,,TRJ20 <CR>
PLOT,TAPE99/BIN=49/J=MIKE20ED/LENGTH=60/TIME=30 <CR>
REPLACE,TAPE99 <CR>

USE,BEDPLT,ED20,TRJ20 <CR>
GRAB,GCS(ZETA,3D) <CR>
GET,BEDPLT,ED20,TRJ20 <CR>
BEDPLT,,,ED20,,TRJ20 <CR>
PLOT,TAPE99/BIN=49/FORMS=WIDE/J=MKMFNLED/LENGTH=60/
     PAYMENT=<UNIV ACCOUNT>/P1=INK/S1=0/TIME=30/
     PLOTTER=SPECIAL <CR>
REPLACE,TAPE99 <CR>

USE,BEDPLT,ED20,ED21,TRJ20,TRJ21 <CR>
GRAB,GCS(ZETA,3D) <CR>
GET,BEDPLT,ED20,ED21,TRJ20,TRJ21 <CR>
BEDPLT,,,ED20,ED21,TRJ20,TRJ21 <CR>
PLOT,TAPE99/BIN=49/J=MKM20-21/LENGTH=60/TIME=30 <CR>
APPEND,TAPE99,TAPE99 <CR>
```

```
      PROGRAM EDPLOT(INPUT,OUTPUT,DATA1,DATA2,TRJ1,TRJ2,TAPE1=INPUT,
     *               TAPE2=OUTPUT,TAPE3=DATA1,TAPE4=DATA2,
     *               TAPE5=TRJ1,TAPE6=TRJ2)
C
C  PROGRAM EDPLOT IS USED TO PLOT PROBE CURRENT OR ELECTRON DENSITY
C  VS. ALTITUDE.  THE FOUR INPUT FILES TO THIS PROGRAM ARE:  DATA1,
C  DATA2, TRJ1, AND TRJ2.  TRJ1 CONTAINS THE TRAJECTORY DATA FOR
C  DATA1.  TRJ2 CONTAINS THE TRAJECTORY DATA FOR DATA2.
C
C  THE FORMATS FOR THESE FILES ARE THE SAME AS THOSE FOR WPROBE AND
C  SWEEP.  NOTE THAT THE DATA1 AND 2 FILES CONTAIN PC OR ED VALUES
C  VERSUS TIME (HENCE THE NEED FOR THE TRAJECTORY DATA).
C
C  WRITTEN BY M K MCINERNEY   DECEMBER, 1979.
C
      REAL AINCR,ALT(6000),ALTI(60),ALTMAX,PC(6000),PCMAX,PCMAX1,
     *PCMAX2,PCMIN,PCMIN1,PCMIN2,PMAX,PMIN,T(60),TIM(1200),XMAX,
     *XMIN,XSCALE,XSIZE,YMAX,YMIN,YSCALE,YSIZE,Z
      INTEGER EOF,FF,FNLED,H,INDEX,KK,LL,MAXI,MAXJ,MM,NN,NO,OVRPLT,
     *PCFILE,PLTED,PLST,PRENDX,TMFILE,TRJCAL,YES
      COMMON/ALL/T,ALTI,TRJCAL,TIM,PC
C
C  INITIALIZE VARIABLES.
```

```
C
      DATA FF/000740334014B/
      DATA PC/6000*1.0/
      DATA NO/1HN/
      DATA YES/1HY/
      PLST=0
      TMFILE=0
      CALL USTART
      CALL DEVICE(XSIZE,YSIZE)
      CALL UMOVE(0.0,100.0)
      CALL UFLUSH
      CALL UALPHA
      PCFILE=-4
C
C CLEAR TERMINAL SCREEN.
C
      WRITE(2,1000) FF
 1000 FORMAT(R6)
      CALL UWAIT(1.5)
C
C FIND OUT IF THIS IS A PLOT OF PROBE CURRENT OR OF ELECTRON DENSITY.
C
      PRINT*, "# ARE YOU PLOTTING PROBE CURRENT OR ELECTRON"
      PRINT*, "  DENSITY?"
      PRINT*, "  "
      PRINT*, "     TYPE 0 FOR PROBE CURRENT."
      PRINT*, "     TYPE 1 FOR ELECTRON DENSITY."
   10 READ*, PLTED
      IF(EOF(1).NE.0.0) GOTO 10
      IF(PLTED.NE.0.AND.PLTED.NE.1) GOTO 10
      IF(PLTED.EQ.0) GOTO 40
      IF(XSIZE.LT.7.60.OR.8.00.LT.XSIZE) GOTO 40
C
C  FIND OUT IF THIS IS A FINAL (10" BY 15" WITH INDIA INK)
C  ELECTRON DENSITY PLOT.
C
      PRINT*, "#  IS THIS A FINAL ELECTRON DENSITY PLOT?"
      PRINT*, "  "
      PRINT*, "   NOTE:  FINAL ELECTRON DENSITY PLOTS ARE TO"
      PRINT*, "          BE PLOTTED OFF LINE ON THE ZETA"
      PRINT*, "          PLOTTER USING INDIA INK."
      PRINT*, "  "
      PRINT*, "   ALSO NOTE:  THERE IS A STANDARD SIZE FOR FINAL"
      PRINT*, "          ELECTRON DENSITY PLOTS; 10 IN. BY 15 IN."
      PRINT*, "          THERE IS ALSO A STANDARD ALTITUDE RANGE"
      PRINT*, "          (50KM-200KM) AS WELL AS A STANDARD ELECTRON"
      PRINT*, "          DENSITY RANGE (10**7 TO 10**12 PER M**3)."
      PRINT*, "   "
   20 PRINT*, "  YES OR NO?"
   30 READ 1100, H
 1100 FORMAT(1A1)
      IF(EOF(1).NE.0.0) GOTO 30
      IF(H.NE.NO.AND.H.NE.YES) GOTO 20
      IF(H.EQ.NO) FNLED=0
```

```
      IF(H.EQ.YES) FNLED=1
      IF(FNLED.EQ.1) CALL UDIMEN(12.0,30.0)
   40 ALTI(1)=-4.0
C
C  FIND OUT WHICH FILE (DATA1 OR DATA2) IS TO BE PLOTTED.
C
      WRITE(2,1000) FF
      CALL UWAIT(1.5)
   50 PRINT*, "* DO YOU WANT TO PLOT DATA FROM DATA1"
      PRINT*, "  OR FROM DATA2?"
      PRINT*, " TYPE 1 FOR DATA1."
      PRINT*, " TYPE 2 FOR DATA2."
   60 READ*, NN
      IF(EOF(1).NE.0.0) GOTO 60
      IF(NN.NE.1.AND.NN.NE.2) GOTO 50
C
C  READ VALUES OF EITHER DATA1 OR DATA2 INTO ARRAY PC.
C
      NN=NN+2
      IF(PCFILE.EQ.NN) GOTO 90
      CALL PREAD(NN,MAXI,MAXJ,PCFILE)
      IF(MAXI.NE.0) GOTO 90
      IF(PCFILE.EQ.-1) GOTO 80
      IF(PLST.EQ.1.AND.PCFILE.GT.0) PCFILE=-4
      IF(NN.EQ.3) GOTO 70
      PRINT*, "* THERE IS NO DATA IN FILE DATA2, TRY A DIFFERENT FILE."
      PCFILE=PCFILE+1
      GOTO 50
   70 PRINT*, "* THERE IS NO DATA IN FILE DATA1, TRY A DIFFERENT FILE."
      PCFILE=PCFILE+1
      GOTO 50
   80 PRINT*, "*  YOU HAVE NOT BEEN ABLE TO FIND ANY DATA"
      PRINT*, "   AFTER FOUR TRIES."
      PRINT*, "  "
      PRINT*, "*  EXECUTION IS TERMINATED."
      IF(PLST.EQ.1) GOTO 500
      STOP 1
C
C  DETERMINE THE TIME IN SECONDS BETWEEN EACH PROBE CURRENT POINT.
C
   90 IF(TMFILE.EQ.NN) GOTO 140
      AINCR=(TIM(2)-TIM(1))/5.0
C
C  ASSIGN A TIME TO EACH PROBE CURRENT.
C
      DO 100 I=1,MAXI
         ALT((I*5)-4)=TIM(I)
         DO 100 J=1,4
            ALT((I*5)-4+J)=TIM(I)+AINCR*J
  100 CONTINUE
C
C  INPUT TRAJECTORY DATA FROM EITHER TAPE5 OR TAPE6.
C  NOTE:  ARRAY LOCATION T(1) IS USED AS A FLAG TO INDICATE IF
C         THE TRAJECTORY DATA HES BEEN READ IN.
```

```
C
      T(1)=0.0
      CALL TREAD(NN+2)
      IF(T(1).NE.0.0) GOTO 120
      PRINT*, "* THERE IS NO DATA IN THE TRAJECTORY FILE"
      IF(NN.EQ.3) PRINT*, "  FOR DATA1."
      IF(NN.EQ.4) PRINT*, "  FOR DATA2."
      PRINT*, "  "
      IF(ALTI(1).NE.-1.0) GOTO 110
      PRINT*, "* YOU HAVE NOT BEEN ABLE TO FIND ANY TRAJECTORY"
      PRINT*, "  DATA AFTER FOUR TRIES."
      PRINT*, "  "
      PRINT*, "* EXECUTION IS TERMINATED."
      IF(PLST.EQ.1) GOTO 500
      STOP 2
  110 IF(NN.EQ.3) PRINT*, "  TRY DATA2."
      IF(NN.EQ.4) PRINT*, "  TRY DATA1."
      ALTI(1)=ALTI(1)+1.0
      GOTO 50
C
C  ASSIGN AN ALTITUDE TO EACH PROBE CURRENT AND DETERMINE THE MINIMUM
C  AND MAXIMUM ALTITUDES OF THE DATA.
C
  120 TRJCAL=0
      ALTMAX=0.0
      DO 130 I=1,MAXJ
         CALL TRAJ(ALT(I),ALT(I))
         ALTMAX=AMAX1(ALTMAX,ALT(I))
         IF(ALTMAX.EQ.ALT(I)) MM=I
  130 CONTINUE
      TMFILE=NN
C
C  MM IS THE INDEX OF THE MAXIMUM ALTITUDE.
C
C  INPUT VALUES OF STARTING AND ENDING ALTITUDES FOR
C  PLOT AND CHECK THEIR VALIDITY.
C
  140 WRITE(2,1000) FF
      CALL UWAIT(1.5)
      PRINT*, "* TO PLOT ASCENDING DATA, HAVE STARTING ALTITUDE"
      PRINT*, "  LESS THAN ENDING ALTITUDE."
      PRINT*, "  "
      PRINT*, "* TO PLOT DESCENDING DATA, HAVE STARTING ALTITUDE"
      PRINT*, "  GREATER THAN ENDING ALTITUDE."
      PRINT*, "  "
      WRITE 1200, ALT(1),ALT(MAXJ)
 1200 FORMAT(1X,"* THE MINIMUM DATA ALTITUDES ARE ",F7.3," AND"/3X,F7.3,
     *" ","  FOR ASCENT AND DESCENT, RESPECTIVELY.")
      PRINT*, "  "
      WRITE 1300, ALTMAX
 1300 FORMAT(" ","* THE MAXIMUM DATA ALTITUDE IS ",F7.3,".")
      PRINT*, "  "
      PRINT*, "* WHAT IS THE STARTING PLOT ALTITUDE? (XXX.XXX)"
  150 READ*, PMIN
```

```
      IF(EOF(1).NE.0.0) GOTO 150
      IF(PMIN.LE.0.0) GOTO 150
      PRINT*, "   "
      PRINT*, "* WHAT IS THE ENDING PLOT ALTITUDE? (XXX.XXX)"
  160 READ*, PMAX
      IF(EOF(1).NE.0.0) GOTO 160
      IF(PMAX.LE.0.0) GOTO 160
C
C   FIND OUT IF THE PREVIOUS PLOT AND THE PRESENT PLOT ARE
C   TO BE SUPERIMPOSED (PLOTTED ON TOP OF EACH OTHER).
C
      IF(PLST.EQ.0) GOTO 180
      IF(XSIZE.LT.7.60.OR.8.00.LT.XSIZE) GOTO 180
      WRITE(2,1000) FF
      CALL UWAIT(1.5)
      PRINT*, "* DO YOU WANT TO HAVE THE PRESENT PLOT AND"
      PRINT*, "  THE PREVIOUS PLOT PLOTTED ON THE SAME"
      PRINT*, "  AXIS ?"
  170 READ 1100, H
      IF(EOF(1).NE.0.0) GOTO 170
      IF(H.NE.NO.AND.H.NE.YES) GOTO 170
      IF(H.EQ.NO) OVRPLT=0
      IF(H.EQ.YES) OVRPLT=1
  180 IF(PMAX.GT.PMIN) GOTO 230
C
C  FOR DESCENT:  DETERMINE THE INDEX OF THE MAXIMUM PLOT ALTITUDE
C                IN THE ARRAY ALT.
C
      DO 190 I=MM,MAXJ
         LL=I
         Z=AMAX1(ALT(I),PMIN)
         IF(Z.EQ.PMIN) GOTO 200
  190 CONTINUE
C
C  DEFINE THE MAXIMUM ALTITUDE ON VERTICAL AXIS
C
  200 YMAX=AINT(ALT(LL))+1.0
      YMAX=AMAX1(YMAX,PMIN)
C
C  FOR DESCENT: DETERMINE THE INDEX OF THE MINIMUM PLOT ALTITUDE
C               IN THE ARRAY ALT.
C
      DO 210 I=LL,MAXJ
         KK=I-1
         Z=AMAX1(ALT(I),PMAX)
         IF(Z.EQ.PMAX) GOTO 220
         IF((KK-LL+1).EQ.2000) GOTO 270
  210 CONTINUE
C
C  DEFINE THE MINIMUM ALTITUDE ON VERTICAL AXIS
C
  220 YMIN=AINT(ALT(KK))
      YMIN=AMIN1(YMIN,PMAX)
      GOTO 290
```

```
C
C FOR ASCENT:   DETERMINE THE INDEX OF THE MINIMUM PLOT ALTITUDE
C              IN THE ARRAY ALT.
C
  230 DO 240 I=1,MM
         LL=I
         Z=AMAX1(ALT(I),PMIN)
         IF(Z.EQ.ALT(I)) GOTO 250
  240 CONTINUE
C
C  DEFINE MINIMUM VALUE ON VERTICAL AXIS.
C
  250 YMIN=AINT(ALT(LL))
      YMIN=AMIN1(YMIN,PMIN)
C
C  FOR ASCENT:  DETERMINE THE INDEX OF THE MAXIMUM PLOT ALTITUDE
C               IN THE ARRAY ALT.
C
      DO 260 I=LL,MM
         KK=I-1
         Z=AMAX1(ALT(I),PMAX)
         IF(Z.EQ.ALT(I)) GOTO 280
         IF((KK-LL+1).EQ.2000) GOTO 270
  260 CONTINUE
      GOTO 280
  270 WRITE 1400, ALT(KK)
 1400 FORMAT(" ","* YOU HAVE ATTEMPTED TO PLOT MORE THAN THE MAXIMUM",/,
     *"  OF 2000 POINTS.  THE ENDING PLOT ALTITUDE HAS BEEN CHANGED",/,
     *"  TO ",F7.3," SO THAT ONLY 2000 POINTS WILL BE PLOTTED.",/,
     *"  PRESS RETURN TO CONTINUE.")
      CALL UPAUSE
      IF(PMAX.LE.PMIN) GOTO 220
C
C  DEFINE THE MAXIMUM PLOT VALUE ON VERTICAL AXIS.
C
  280 YMAX=AINT(ALT(KK))+1.0
      YMAX=AMAX1(YMAX,PMAX)
C
C  LL IS THE INDEX OF THE STARTING ALTITUDE.
C  KK IS THE INDEX OF THE ENDING ALTITUDE.
C
  290 IF((KK-LL).GE.4) GOTO 300
      PRINT*, "* YOU MUST PLOT AT LEAST 5 DATA POINTS.  TRY AGAIN."
      PRINT*, " "
      PRINT*, "  PRESS RETURN TO CONTINUE."
      CALL UPAUSE
      GOTO 140
C
C  FIND THE MINIMUM AND MAXIMUM VALUES OF PROBE CURRENT OVER THE
C  INTERVAL TO BE PLOTTED.
C
  300 IF(OVRPLT.EQ.1) GOTO 430
      PCMIN=1.0E100
      PCMAX=0.0
```

```
      DO 310 I=LL,KK
         IF(PC(I).EQ.1.0) GOTO 310
         IF(PC(I).EQ.2.0) GOTO 310
         PCMIN=AMIN1(PCMIN,PC(I))
         PCMAX=AMAX1(PCMAX,PC(I))
         PRENDX=I
  310 CONTINUE
C
C  DETERMINE MIN AND MAX PROBE CURRENT VALUES AS POWERS OF 10.
C
      PCMIN2=ALOG10(PCMIN)
      PCMAX2=ALOG10(PCMAX)
      IF(PCMIN2.LT.0.0) PCMIN2=AINT(PCMIN2)-1.0
      IF(PCMIN2.GE.0.0) PCMIN2=AINT(PCMIN2)
      IF(PCMAX2.LT.0.0) PCMAX2=AINT(PCMAX2)
      IF(PCMAX2.GE.0.0) PCMAX2=AINT(PCMAX2)+1.0
      IF(FNLED.EQ.1) GOTO 400
C
C  FIND OUT IF THE ED/PC AXIS IS TO BE SELF-SCALED.
C
      WRITE(2,1000) FF
      CALL UWAIT(1.5)
      PRINT*, "* DO YOU WANT TO SCALE THE PROBE CURRENT/"
      PRINT*, "  ELECTRON DENSITY AXIS YOURSELF OR DO YOU"
      PRINT*, "  WISH TO HAVE THE PLOT ROUTINE SCALE IT?"
      PRINT*, " TYPE YES TO SCALE YOURSELF."
      PRINT*, " TYPE NO TO HAVE PLOT ROUTINE SCALE."
  320 READ 1100, H
      IF(EOF(1).NE.0.0) GOTO 320
      IF(H.EQ.NO) GOTO 400
      IF(H.NE.YES) GOTO 320
  330 PRINT*, "* WHAT IS THE MINIMUM CURRENT AXIS VALUE"
      PRINT*, "  AS A POWER OF 10,I.E. 10**X."
      PRINT*, "  ENTER X."
  340 READ*, PCMIN1
      IF(EOF(1).NE.0.0) GOTO 340
      IF(PCMIN1.LE.PCMIN2) GOTO 360
      PRINT*, "* WARNING *  THE MINIMUM CURRENT VALUE IS ",PCMIN
      PRINT*, "  DO YOU WISH TO CONTINUE ANYWAY?"
  350 READ 1100, H
      IF(EOF(1).NE.0.0) GOTO 350
      IF(H.EQ.NO) GOTO 330
      IF(H.EQ.YES) GOTO 360
      GOTO 350
  360 PRINT*, "* ENTER THE MAXIMUM CURRENT AXIS VALUE"
      PRINT*, "  AS A POWER OF 10, I.E. 10**X."
  370 READ*, PCMAX1
      IF(EOF(1).NE.0.0) GOTO 370
      IF(PCMAX1.GE.PCMAX2) GOTO 390
      PRINT*, "* WARNING *  THE MAXIMUM CURRENT VALUE IS ",PCMAX
      PRINT*, "  DO YOU WISH TO CONTINUE ANYWAY?"
  380 READ 1100, H
      IF(EOF(1).NE.0.0) GOTO 380
      IF(H.EQ.YES) GOTO 390
```

```
      IF(H.EQ.NO) GOTO 360
      GOTO 380
  390 IF(PCMAX1.LE.PCMIN1) GOTO 330
      PCMIN2=PCMIN1
      PCMAX2=PCMAX1
  400 XMIN=10**PCMIN2
      XMAX=10**PCMAX2
      XSCALE=(XMAX-XMIN)/10.0
      YSCALE=(YMAX-YMIN)/10.0
C
C  PLOT.
C
      CALL URESET
      CALL UERASE
      PLST=1
      IF(FNLED.NE.1) GOTO 410
      XMIN=1.0E07
      XMAX=1.0E12
      YMIN=50.0
      YMAX=200.0
      CALL UDAREA(0.0,10.05875,5.0,20.10375)
      CALL USET("NOXLABEL")
      CALL USET("NOYLABEL")
      CALL USET("TICAXES")
      GOTO 420
  410 IF(XSIZE.LT.7.6.OR.8.0.LT.XSIZE) CALL UDAREA(0.0,XSIZE,0.0,YSIZE)
      IF(7.6.LT.XSIZE.AND.XSIZE.LT.8.0) CALL UDAREA(.25,XSIZE,.25,YSIZE)
      CALL USET("XBOTHLABELS")
      CALL USET("YBOTHLABELS")
      CALL UPSET("YLABEL","ALTITUDE;")
      IF(PLTED.EQ.0) CALL UPSET("XLABEL","PROBE CURRENT;")
      IF(PLTED.NE.0) CALL UPSET("XLABEL","ELECTRON DENSITY;")
  420 CALL USET("LOGXAXIS")
      CALL USET("XLOGARITHMIC")
      CALL USET("OWNSCALE")
      CALL UPSET("TICY",5.0)
      CALL UWINDO(XMIN,XMAX,YMIN,YMAX)
      CALL UAXIS(XMIN,XMAX,YMIN,YMAX)
  430 INDEX=LL
      PRENDX=INDEX
      IF(PC(INDEX).EQ.1.0.OR.PC(INDEX).EQ.2.0) GOTO 450
      CALL UMOVE(PC(INDEX),ALT(INDEX))
      PRENDX=INDEX
      INDEX=INDEX+1
      IF(INDEX.GT.KK) GOTO 480
  440 IF(PC(INDEX).EQ.1.0.OR.PC(INDEX).EQ.2.0) GOTO 450
      CALL UPEN(PC(INDEX),ALT(INDEX))
      PRENDX=INDEX
      INDEX=INDEX+1
      IF(INDEX.GT.KK) GOTO 480
      GOTO 440
  450 CALL UMOVE(PC(PRENDX),ALT(INDEX))
      IF(PC(INDEX+1).NE.1.0.AND.PC(INDEX+1).NE.2.0) GOTO 460
      INDEX=INDEX+1
```

```
      IF(INDEX.GT.KK) GOTO 480
      GOTO 450
  460 INDEX=INDEX+1
      IF(INDEX.GT.KK) GOTO 480
      IF(PC(INDEX+1).EQ.1.0.OR.PC(INDEX+1).EQ.2.0) GOTO 470
      CALL UMOVE(PC(INDEX),ALT(INDEX))
      PRENDX=INDEX
      INDEX=INDEX+1
      IF(INDEX.GT.KK) GOTO 480
      GOTO 440
  470 CALL USET("NSYMBOL")
      CALL UPSET("SZMARKER",.08)
      CALL UPSET("SYMBOL",(2.0*FLOAT(PCFILE)))
      CALL UPEN(PC(INDEX),ALT(INDEX))
      CALL USET("LINE")
      PRENDX=INDEX
      INDEX=INDEX+1
      IF(INDEX.GT.KK) GOTO 480
      GOTO 450
  480 CALL UBELL
      CALL UHOME
      CALL UFLUSH
      CALL UPAUSE
C
C  FIND OUT IF ANOTHER PLOT IS TO BE MADE.
C
      CALL UALPHA
      WRITE(2,1000) FF
      CALL UWAIT(2.5)
      PRINT*, "* DO YOU WISH TO PLOT AGAIN WITH DIFFERENT PARAMETERS?"
  490 READ 1100, H
      IF(EOF(1).NE.0.0) GOTO 490
      IF(H.EQ.YES) GOTO 40
      IF(H.EQ.NO) GOTO 500
      GOTO 490
  500 CALL UEND
      WRITE(2,1000) FF
      CALL UWAIT(1.5)
      IF(XSIZE.LT.7.60.OR.8.00.LT.XSIZE) GOTO 510
C
C  PRINT FINAL MESSAGE FOR ZETA PLOTS.
C
      PRINT*, "* PLOT OUTPUT IS IN LOCAL FILE TAPE99."
      PRINT*, "  REMEMBER TO SUBMIT IT TO BE PLOTTED."
      PRINT*, "  "
      PRINT*, "* FOR AN ONLINE ZETA PLOT DONE IN BLACK INK"
      PRINT*, "  WITH A ROLLING WRITER PEN, THE FOLLOWING"
      PRINT*, "  COMMAND CAN BE USED;"
      PRINT*, "  "
      PRINT*, "  PLOT,TAPE99/BIN=49/J=MIKE7/LENGTH=60/TIME=30."
      PRINT*, "  "
      PRINT*, "  FOR AN OFFLINE PLOT USING WIDE PAPER AND INDIA"
      PRINT*, "  INK, THE FOLLOWING COMMAND CAN BE USED;"
      PRINT*, "  "
```

```
      PRINT*, "  PLOT,TAPE99/BIN=49/FORMS=WIDE/J=MIKE7/LENGTH=60/"
      PRINT*, "        PAYMENT=<UNIV ACCOUNT>/P1=INK/S1=0/TIME=30/"
      PRINT*, "        PLOTTER=SPECIAL."
      PRINT*, "  "
  510 STOP
      END
      SUBROUTINE TRAJ(TIME,HEIGHT)
C
C COMPUTES INTERPOLATED ALTITUDE VALUE FOR ANY TIME AFTER LAUNCH.
C MODIFIED FOR USE WITH PROGRAM SWEEP. ALTITUDE AND TIME VALUES
C AT 10 SECOND INTERVALS MUST BE PROVIDED VIA ARRAYS T AND ALTI
C FOR ENTIRE PERIOD OF FLIGHT. SWEEP READS THESE ARRAYS FROM FILE
C TRJTRY.
C INPUT  ARGUMENT:   TIME = TIME IN SECONDS AFTER LAUNCH.
C OUTPUT ARGUMENT: HEIGHT = ALTITUDE IN KILOMETERS.
C
      REAL ALTI(60),PC(6000),T(60),TIM(1200)
      INTEGER INCR,TRJCAL
      COMMON/ALL/T,ALTI,TRJCAL,TIM,PC
      INCR=0
C
C IF FIRST CALL TO TRAJ, INITIALIZE VARIABLES. IF NOT SKIP TO 10.
C
      IF(TRJCAL.GT.0) GOTO 20
      I=4
      TRJCAL=1
   10 INCR=1
      IM1=I-1
      IM2=I-2
   20 IF(TIME.LT.T(IM2)) WRITE(2,1000) TIME
 1000 FORMAT(" "/"TIME LESS THAN LOWEST TRAJECTORY POINT,  TIME=",F7.3)
C
C LOCATE TIME VALUES WHICH BRACKET PRESENT INPUT TIME VALUE.
C
      IF(TIME.LE.T(I)) GOTO 30
      I=I+1
      GOTO 10
C
C IF PRESENT TIME VALUE IS IN SAME INTERVAL AS PREVIOUS ONE,
C COMPUTE ALTITUDE WITH OLD COEFFICIENTS. IF NOT, COMPUTE NEW
C COEFFICIENTS FIRST.
C
   30 IF(INCR.EQ.1) GOTO 50
   40 HEIGHT=A*TIME*TIME+B*TIME+C
      RETURN
   50 BRAC1=(T(I)-T(IM1))*(ALTI(IM1)-ALTI(IM2))
      BRAC2=(T(IM1)-T(IM2))*(ALTI(I)-ALTI(IM1))
      TOP=BRAC1-BRAC2
      BRAC1=(T(IM1)-T(IM2))*(T(I)*T(I)-T(IM1)*T(IM1))
      BRAC2=(T(I)-T(IM1))*(T(IM1)*T(IM1)-T(IM2)*T(IM2))
      BOTTOM=BRAC2-BRAC1
      A=TOP/BOTTOM
      B=(ALTI(IM1)-ALTI(IM2))-A*(T(IM1)*T(IM1)-T(IM2)*T(IM2))
      B=B/(T(IM1)-T(IM2))
```

```
      C=ALTI(IM2)-A*T(IM2)*T(IM2)-B*T(IM2)
      GOTO 40
      END
      SUBROUTINE PREAD(NN,MAXI,MAXJ,PCFILE)
C
C  THIS SUBROUTINE READS DATA FROM EITHER TAPE3 OR
C  TAPE4 INTO ARRAYS TIM AND PC.
C
      REAL ALTI(60),PC(6000),T(60),TIM(1200)
      INTEGER EOF,MAXI,MAXJ,NN,PCFILE,TRJCAL
      COMMON/ALL/T,ALTI,TRJCAL,TIM,PC
      REWIND NN
      I=0
      J=0
   10 I=I+1
      J=J+5
      JL=J-4
      READ(NN,1000) TIM(I),(PC(K),K=JL,J)
      IF(EOF(NN)) 20,10
 1000 FORMAT(F7.1,5E13.4)
   20 MAXI=I-1
      MAXJ=J-5
      IF(MAXI.NE.0) PCFILE=NN
      RETURN
      END
      SUBROUTINE DEVICE(MAXXDIM,MAXYDIM)
C
C  THIS SUBROUTINE SETS THE PLOT SIZE (IN INCHES)
C  DEPENDING ON WHICH DEVICE THE PLOT IS BEING
C  MADE.
C
      REAL LIMIT(8),MAXXDIM,MAXYDIM
      CALL USTUD(LIMIT)
      IF(LIMIT(6).LT.6.0) GOTO 10
      IF(LIMIT(6).LT.8.0) GOTO 20
      IF(LIMIT(6).LT.11.0) GOTO 30
      IF(LIMIT(6).LT.14.0) GOTO 40
      IF(LIMIT(6).LT.15.0) GOTO 50
C
C   DEVICE IS ALPH OR ADDR.
C
   10 MAXXDIM=7.09
      MAXYDIM=5.74
      RETURN
C
C   DEVICE IS TEKT.
C
   20 MAXXDIM=7.49
      MAXYDIM=5.71
      RETURN
C
C   DEVICE IS CALC OR ZETA.
C
   30 MAXXDIM=7.75
```

```
      MAXYDIM=9.75
      CALL UDIMEN(8.5,11.0)
      RETURN
C
C   DEVICE IS PRNT.
C
   40 MAXXDIM=12.99
      MAXYDIM=7.37
      RETURN
C
C   DEVICE IS TK14.
C
   50 MAXXDIM=14.33
      MAXYDIM=10.91
      RETURN
      END
      SUBROUTINE TREAD(NUNIT)
C
C  THIS SUBROUTINE READS IN THE TRAJECTORY DATA FROM
C  EITHER TAPE5 OR TAPE6 INTO ARRAYS T AND ALTI.
C
      REAL ALTI(60),PC(6000),T(60),TIM(1200)
      INTEGER EOF,NUNIT,TRJCAL
      COMMON/ALL/T,ALTI,TRJCAL,TIM,PC
      REWIND NUNIT
      I=1
   10 I=I+1
      READ(NUNIT,1000) T(I),ALTI(I)
      IF(EOF(NUNIT)) 20,10
 1000 FORMAT(F7.1,F10.3)
   20 IF(I.NE.2) T(1)=FLOAT(NUNIT)
      RETURN
      END
```

APPENDIX XIII. Listing of IBM Fortran IV program EEDPROC.

The JCL and input parameters to run this program using energetic particle data from flight 18.1021 are included. Data from T+40 sec through T+350 sec was analyzed.

Parameter ICHECK sets the diagnostic level. A value between 0 and 3 may be enterd, with 0 resulting in no diagnostics and 3 giving extensive diagnostics. A value of 1 is usually used.

The last two input values, 3600 and 350 in this example, are the count calibration levels for 15 counts and 0 counts, respectively. Each of the 16 count levels associated with each detector has a corresponding digital value between 0 and 4095. The digital-to-analog converter on the payload assigns an analog voltage between 0 and 5 V to each count level. When the analog voltage is digitized a digital value between 0 and 4095 is then assigned to each count level.

The count level calibration is performed by running program EEDPROC over a selected time interval with ICHECK=3. At a diagnostic level of 3 the data in each record between FSTSEC and LSTSEC will be printed. The digital values associated with each count level can then be obtained from the printed output.

Since the count levels have the same voltage (and therefore digital) spacing the digital values associated with the intermediate count levels are computed within program EEDPROC by linearly interpolating between the maximum and minimum count levels.

Program EEDPROC also has the capability of calculating the rocket's angle with respect to the Earth's magnetic field. Subroutine ZANGLE uses the magnetometer signal to perform this calculation. The amplitude of the magnetometer signal is proportional to its orientation along a magnetic field line: a maximum when parallel and a minimum when anti-parallel. The rocket's azimuthal position at a certain time can be determined by comparing the magnetometer amplitude at that time to the local minimum and maximum.

The azimuth angle calculation was not done for the eclipse flights. The subroutine ZANGLE call statement (four lines above the statement labeled 907) has been disabled by changing it to a comment statement.

```
//FRANK21 JOB
/*ID PS=1968
/*ID CODE=
/*ID NAME='VOSS',PRINT=LOCAL,BIN=09
/*ID LINES=25000,TIME=20,IOREQ=8000,REGION=200K
/*ID CARDS=2000
/*SETUP UNIT=TAPE,R=C222,ID=AER018***NORING
// EXEC FORTLDGO,REGION.GO=200K
//FORT.SYSIN DD *
C     PROGRAM EEDPROC
C     READS WALLOPS DIGITAL TAPES AND FORMATS EED DATA SAMPLES FOR
```

```
C     ANALYSIS BY SUBROUTINES
C     WRITTEN BY H D VOSS  1975
C     MODIFIED BY F M BRASWELL  1979
       INTEGER TI(1000)
      INTEGER*4 TIMPRT,RECHRS,RECMIN,EEDCH1,EEDCH2,EEDCH3,EEDCH4,EEDCH5,
     1CURHRS,CURMIN,HRS,FSTSEC,DATAID(20)
      INTEGER*2 ARRAY(2008),EED1(5000),EED2(5000),EED3(5000),EED4(5000),
     1EED5(5000),COUNT1(2000),COUNT2(2000),COUNT3(2000),COUNT4(2000),
     1COUNT5(400)
      DOUBLE PRECISION RECTIM,ENDREC,ENDNEX,CVTTIM,INITIM
       DOUBLE PRECISION ANGLE(1000)
      DOUBLE PRECISION STARTM,ENDTIM,CURTIM,LCHTIM
      REAL*4 ITINCR
      COMMON ARRAY,TIMPRT,NUMREC,NUMHED,NUMCAL,RECHRS,RECMIN,RECSEC
      NAMELIST/PARMS/NUMHED,NUMCAL,ICHECK
      NAMELIST/CHANS/EEDCH1,EEDCH2,EEDCH3,EEDCH4,EEDCH5
      NAMELIST/TIMES/FSTSEC,LSTSEC
      CVTTIM(HRS,MINS,SECS)=3.6D3*HRS+6.D1*MINS+1.D0*SECS
      RATE = 25000.
      IRATE = IFIX(RATE)
      READ (5,PARMS)
      READ (5,TIMES)
      READ (5,CHANS)
      READ (5,801) LCHRS,LCHMIN,LCHSEC
  801 FORMAT (2I2,1X,I2)
      READ (5,802) DATAID
  802 FORMAT (20A4)
      WRITE (11,807)
      WRITE (6,807)
  807 FORMAT (1H1)
      WRITE (11,PARMS)
      WRITE (11,TIMES)
      WRITE (11,CHANS)
      WRITE (6,804) DATAID
  804 FORMAT (1X,20A4)
      WRITE (6,803) LCHRS,LCHMIN,LCHSEC
  803 FORMAT (1X,'LAUNCH TIME:   ',2I2,':',I2)
      IWRITE=0
      NUMREC=0
      TIMPRT=1
      READ (5,489) ICMAX,ICMIN
  489 FORMAT (2I6)
      WRITE (6,486) ICMAX,ICMIN
  486 FORMAT (/,1X,2I6)
      IDELCO=(ICMAX-ICMIN)/16
      K=1
      DO 589 J=1,2000
      COUNT1(J)=0
      COUNT2(J)=0
      COUNT3(J)=0
      COUNT4(J)=0
  589 COUNT5(J)=0
      CALL TPREAD
C   CALCULATE TIME OF FIRST DATA SAMPLE NEEDED
```

```
      RLCSEC=LCHSEC
      LCHTIM=CVTTIM(LCHRS,LCHMIN,RLCSEC)
      WRITE (11,850) LCHTIM
  850 FORMAT (1X,'LAUNCH TIME CONVERTED TO SECONDS:  ',F9.2/)
      STARTM=LCHTIM+FSTSEC
      ENDTIM=LCHTIM+LSTSEC
      CURTIM=STARTM
      INITIM=STARTM
C  LOCATE RECORD CONTAINING FIRST SAMPLE NEEDED
  130 CALL DAREAD (IWRITE)
      RECTIM=CVTTIM(RECHRS,RECMIN,RECSEC)
C***  IF(ICHECK.GE.1) WRITE (11,813) RECTIM
C*813 FORMAT ('+',50X,'RECTIM=   ',F12.5)
      ENDREC=RECTIM + 2000./RATE
      IF (RECTIM-INITIM) 133,133,132
  132 WRITE (11,812)
  812 FORMAT (1X,'START TIME IS EARLIER THAN TIME OF FIRST RECORD')
      STOP 5
  133 IF (INITIM.LE.ENDREC) GO TO 135
      ENDNEX=RECTIM + 4000./RATE
      IF (INITIM.LE.ENDNEX.AND.ICHECK.GE.2) IWRITE=1
      GO TO 130
C  DETERMINE SUBSCRIPT OF FIRST DATA SAMPLES NEEDED
  135 ITINCR=INITIM-RECTIM
      INDEX=RATE*ITINCR+6.
      IDXFST=5+5*((INDEX-1)/5)
      IF (IDXFST.LT.2005) GO TO 136
      CALL DAREAD (IWRITE)
      RECTIM=CVTTIM(RECHRS,RECMIN,RECSEC)
      IDXFST=5
C  DETERMINE SUBSCRIPTS OF FIRST SAMPLES FOR EACH CHANNEL
  136 IDXCH1 = IDXFST + EEDCH1
      IDXCH2 = IDXFST + EEDCH2
      IDXCH3 = IDXFST + EEDCH3
      IDXCH4 = IDXFST + EEDCH4
      IDXCH5 = IDXFST + EEDCH5
      IF(ICHECK.GE.1) WRITE(11,805) INDEX,IDXFST,IDXCH1,IDXCH2,IDXCH3,
     1IDXCH4,IDXCH5,NUMREC
  805 FORMAT (1X,'INDEX=',I4,3X,'IDXFST=',I4,3X,'IDXCH1=',I4,3X,'IDXCH2=
     1',I4,3X,'IDXCH3=',I4,3X,'IDXCH4=',I4,3X,'IDXCH5=',I4,3X,
     2'FIRST RECORD NUMBER= ',I4)
      IWRITE=0
      I=1
  137 EED1(I) = ARRAY(IDXCH1)
      EED2(I) = ARRAY(IDXCH2)
      EED3(I) = ARRAY(IDXCH3)
      EED4(I) = ARRAY(IDXCH4)
      EED5(I) = ARRAY(IDXCH5)
      IDXCH1 = IDXCH1 + 5
      IF(IDXCH1.GT.2005) GO TO 140
      IDXCH2=IDXCH2+5
      IDXCH3=IDXCH3+5
      IDXCH4=IDXCH4+5
      IDXCH5=IDXCH5+5
```

```
      GO TO 150
  140 CALL DAREAD (IWRITE)
      RECTIM=CVTTIM(RECHRS,RECMIN,RECSEC)
      IDXCH1=EEDCH1 + 5
      IDXCH2=EEDCH2 + 5
      IDXCH3=EEDCH3 + 5
      IDXCH4=EEDCH4 + 5
      IDXCH5=EEDCH5 + 5
  150 I = I + 1
      IF (I.LE.(IRATE/5)) GO TO 137
       DO 123 I=1,1000
       TI(I)=0
123    CONTINUE
       DO 1234 I=1,1000,20
       TI(I)=I
1234   CONTINUE
      IF (ICHECK.GE.1) WRITE (11,814) IDXCH1,IDXCH2,IDXCH3,IDXCH4,IDXCH5
     1,NUMREC
  814 FORMAT(1X,'END INDICES:  IDXCH1=',I4,3X,'IDXCH2=',I4,3X,'IDXCH3=',
     1I4,3X,'IDXCH4=',I4,3X,'IDXCH5=',I4,3X,'LAST RECORD NUMBER=',I4)
      IF (ICHECK.GE.3) WRITE (11,816) NUMREC
  816 FORMAT (//1X,'RECORD NUMBER: ',I4)
      IF (ICHECK.GE.3) WRITE (11,815) ARRAY
  815 FORMAT (/101(' ',20I6/))
      CALL UTIME (CURTIM,THRS,TMINS,CURSEC)
      CURHRS=THRS
      CURMIN=TMINS
      IF(ICHECK .GT. 1)GOTO 907
      M = IDINT(CURTIM  - LCHTIM)
      IF(MOD(M,40) .NE. 0)GOTO 310
C***   CALL ZANGLE(EED5,ANGLE,TI)
      JARMAX=IRATE/100
      XICMIN=FLOAT(ICMIN)
      XIDELC=FLOAT(IDELCO)
907   WRITE (6,809) CURHRS, CURMIN,CURSEC
      WRITE (6,808)
      CALL NICEPR(EED1,JARMAX,XICMIN,XIDELC)
  809 FORMAT (1H1,'TIME OF PRESENT DATA BLOCK: ',2I2,':',F5.2/)
  808 FORMAT (/1X,'EED CHANNEL 1 DATA SAMPLES'//)
      WRITE (6,809) CURHRS, CURMIN,CURSEC
      WRITE (6,810)
      CALL NICEPR(EED2,JARMAX,XICMIN,XIDELC)
  810 FORMAT (/1X,'EPS CHANNEL 2 DATA SAMPLES'//)
      WRITE (6,809) CURHRS, CURMIN,CURSEC
      WRITE (6,811)
      CALL NICEPR(EED3,JARMAX,XICMIN,XIDELC)
  811 FORMAT (/1X,'EPS CHANNEL 3 DATA SAMPLES'//)
      WRITE (6,809) CURHRS, CURMIN,CURSEC
      WRITE (6,817)
      CALL NICEPR(EED4,JARMAX,XICMIN,XIDELC)
  817 FORMAT (/1X,'EPS CHANNEL 4 DATA SAMPLES'//)
      WRITE (6,809) CURHRS, CURMIN,CURSEC
      WRITE (6,818)
      CALL NICEPR(EED5,JARMAX,XICMIN,XIDELC)
```

```
818   FORMAT(/1X,'EPS CHANNEL 5 DATA SAMPLES'//)
  310 IRAMPS=0
      K7 = 0
503   I=0
  500 I=I+1
      IF (I-999) 501,504,504
  501 IF((EED1(I)-EED1(I+1))-( 4*IDELCO)) 500,500,502
  502 IRAMPS=IRAMPS+1
      I=I+50
      IF (I-999) 501,504,504
  504 COUNT1(K)=IRAMPS*16-(EED1(1)-ICMIN)/IDELCO+(EED1(1000)-ICMIN)/
     1IDELCO
      IRAMPS=0
      I=1
  505 KA=EED2(I)-EED2(I+1)
      KB=EED2(I)-EED2(I+2)
      IF (KA.GT. 5*IDELCO .OR.KB.GT. 6*IDELCO) GO TO 507
      I=I+1
      IF (I-998)505,510,510
  507 IRAMPS=IRAMPS+1
      I=I+50
      IF (I-998) 505,510,510
  510 COUNT2(K)=IRAMPS*16-(EED2(1)-ICMIN)/IDELCO+(EED2(1000)-ICMIN)/
     1IDELCO
      IRAMPS=0
      I=1
  511 KA=EED3(I)-EED3(I+1)
      KB=EED3(I)-EED3(I+2)
      IF (KA.GT. 4*IDELCO .OR.KB.GT. 5*IDELCO) GO TO 512
      I=I+1
      IF (I-998) 511,515,515
  512 IRAMPS=IRAMPS+1
      I=I+50
      IF (I-998) 511,515,515
  515 COUNT3(K)=IRAMPS*16-(EED3(1)-ICMIN)/IDELCO+(EED3(1000)-ICMIN)/
     1IDELCO
      IRAMPS=0
      I=1
  516 KA=EED4(I)-EED4(I+1)
      KB=EED4(I)-EED4(I+2)
      IF (KA.GT. 3*IDELCO .OR.KB.GT. 4*IDELCO) GO TO 517
      I=I+1
      IF (I-998)516,520,520
  517 IRAMPS=IRAMPS+1
      I=I+60
      IF (I-998)516,520,520
  520 COUNT4(K)=IRAMPS*16-(EED4(1)-ICMIN)/IDELCO+(EED4(1000)-ICMIN)/
     1IDELCO
      IRAMPS=0
      I=1
  521 KA=EED5(I)-EED5(I+1)
      KB=EED5(I)-EED5(I+2)
      IF (KA.GT. 3*IDELCO .OR.KB.GT. 4*IDELCO) GO TO 522
      I=I+1
```

```
      IF (I-998)521,525,525
  522 IRAMPS=IRAMPS+1
      I=I+100
      IF (I-998)521,525,525
  525 COUNT5(K)=IRAMPS*16-(EED5(1)-ICMIN)/IDELCO+(EED5(1000)-ICMIN)/
     1IDELCO
      K=K+1
      K7=K7+1000
      IF(K7 .GT. 4500)GOTO 776
      DO 530 J7=1,1000
      EED1(J7)=EED1(J7+K7)
      EED2(J7)=EED2(J7+K7)
      EED3(J7)=EED3(J7+K7)
      EED4(J7)=EED4(J7+K7)
530   EED5(J7)=EED5(J7+K7)
      GOTO 503
776   CURTIM=CURTIM+1.D0
      IF (CURTIM.GT.ENDTIM) GO TO 533
      INITIM=CURTIM
      IF (ICHECK.GE.3) IWRITE=1
      GOTO 135
  533 WRITE (6,488) COUNT1
      WRITE (6,488) COUNT2
      WRITE (6,488) COUNT3
      WRITE (6,488) COUNT4
      WRITE (6,488) COUNT5
  488 FORMAT (/,1X,'COUNTS PER SECOND',//,(' ',20I6))
      DO 717 J=1,K
      M=FSTSEC+J-1
      WRITE(7,719)COUNT1(J),COUNT2(J),COUNT3(J),COUNT4(J),
     1COUNT5(J),M
719   FORMAT(6I6)
717   CONTINUE
      STOP 1
      END
      SUBROUTINE UTIME (TIM,HRS,MINS,SECS)
      REAL*4 MINS
      TI1=TIM/3600.
      HRS=AINT(TI1)
      REM1=AMOD(TIM,3600.)
      ATI2=REM1/60.
      MINS=AINT(ATI2)
      SECS=AMOD(REM1,60.)
      RETURN
      END
      SUBROUTINE TPREAD
C   READS RECORDS FROM WALLOPS FARADAY ROTATION DATA TAPES.
      INTEGER*2 ARRAY(2008)
      INTEGER*4 RECHRS,RECMIN
      COMMON ARRAY,TIMPRT,NUMREC,NUMHED,NUMCAL,RECHRS,RECMIN,RECSEC
C   SKIP HEADER RECORDS
      DO 100 L=1,NUMHED
      READ (12,801,ERR=1000,END=2000) (ARRAY(M),M=1,10)
  801 FORMAT (10A2)
```

```
      NUMREC=NUMREC+1
      WRITE (11,807)
  807 FORMAT (1X,'HEADER RECORD READ')
  100 CONTINUE
C  SKIP CALIBRATION RECORDS
      DO 110 L=1,NUMCAL
      READ (12,802,ERR=1000,END=2000) (ARRAY(M),M=1,1005)
  802 FORMAT (5(201A2))
      WRITE (11,808)
  808 FORMAT (1X,'CAL RECORD READ')
  110 NUMREC=NUMREC+1
      RETURN
      ENTRY DAREAD (IWRITE)
C  READ DATA RECORD
      READ (12,803,ERR=1000,END=2000) ARRAY
  803 FORMAT (8(251A2))
      NUMREC=NUMREC+1
C***  WRITE (11,809) NUMREC
C*809 FORMAT (1X,'DATA RECORD',I4,' READ')
      CALL GETIME
      IF (IWRITE.NE.1) RETURN
      WRITE (11,805) NUMREC
  805 FORMAT (//1X,'RECORD NUMBER: ',I4)
      WRITE (11,806) ARRAY
  806 FORMAT (/101(' ',20I6/))
      RETURN
 1000 WRITE (11,1010)
 1010 FORMAT (1X,'*TPREAD*   TAPE ERROR')
      STOP 2
 2000 WRITE (11,2010)
 2010 FORMAT (1X,'*TPREAD*   END OF TAPE FILE')
      STOP 3
      END
      SUBROUTINE GETIME
      REAL MULT,MSECS
      COMMON I,TIME,NUMREC,NUMHED,NUMCAL,RECHRS,RECMIN,RECSEC
      INTEGER*4 RECHRS,RECMIN,TIME
      INTEGER*2 I(2008),K(16)
      IWDNUM=2006
   20 ITIME=I(IWDNUM)
      IST=1
      IF(ITIME.GE.0) GO TO 30
      K(16)=1
      ITIME=ITIME+32768
      IST=2
   30 DO 50 M=IST,16
      N=17-M
      ITEST=2**(N-1)
      IF(ITIME.GE.ITEST) GO TO 70
      K(N)=0
      GO TO 50
   70 K(N)=1
      ITIME=ITIME-ITEST
   50 CONTINUE
```

```
      IF(ITIME.NE.0) WRITE(6,800)
  800 FORMAT (1X,'ERROR IN READING TIME WORD. RESULT NOT ZERO')
      IF(IWDNUM-2007) 100,200,300
  100 MULT=0.0001
      MSECS=0.0
      DO 110 IDX=1,13,4
      ITMSEC=8*K(IDX+3)+4*K(IDX+2)+2*K(IDX+1)+K(IDX)
      MSECS=MSECS+(ITMSEC*MULT)
      MULT=MULT*10.
  110 CONTINUE
      IWDNUM=2007
      GO TO 20
  200 RECSEC=40*K(7)+20*K(6)+10*K(5)+8*K(4)+4*K(3)+2*K(2)+K(1)+MSECS
      RECMIN=40*K(15)+20*K(14)+10*K(13)+8*K(12)+4*K(11)+2*K(10)+K(9)
      IWDNUM=2008
      GO TO 20
  300 RECHRS=20*K(6)+10*K(5)+8*K(4)+4*K(3)+2*K(2)+K(1)
C**** IF (TIME.GE.1) WRITE (11,805) RECHRS,RECMIN,RECSEC
C*805 FORMAT ('+',28X,'TIME: ',I2,I2,':',F7.4)
      RETURN
      END
        SUBROUTINE ZANGLE(EED5,ANGLE,TI)
C     THIS SUBROUTINE FINDS THE ROCKET AZIMUTH ANGLE
C      AT A GIVEN PARTICULAR TIME
        INTEGER*2 EED5(1000), MAX,MIN
       INTEGER*2 MAX1,MIN1,MAX2,MIN2
       INTEGER TI(1000),J,M,I,P,L,N
       DOUBLE PRECISION VAL,DC,A,ANGLE(1000),DARSIN,NORL,DABS,K
       DOUBLE PRECISION ED(1000)
       ED(1)=EED5(1)
       ED(2)=EED5(1)+EED5(2)+EED5(3)
       ED(2)=ED(2)/3
       DO 95 I=3,998,1
       ED(I)=EED5(I-2)+EED5(I-1)+EED5(I)+EED5(I+1)+EED5(I+2)
       ED(I)=ED(I)/5
95     CONTINUE
       ED(999)=EED5(998)+EED5(999)+EED5(1000)
       ED(999)=ED(999)/3
       ED(1000)=EED5(1000)
C      TO FIND THE MAX AND MIN OF EED5
       MAX1=EED5(1)
       MIN1=EED5(1)
       DO 1000 M=2,1000
       K=EED5(M)-MAX1
       K=DABS(K)
       IF(K .GE.100) GO TO 2000
       IF (EED5(M) .GT. MAX1) GO TO 1001
       IF (EED5(M) .GE. MIN1) GO TO 1000
       MIN1=EED5(M)
       GO TO 1000
1001   MAX1=EED5(M)
1000   CONTINUE
       GO TO 3000
2000   CONTINUE
```

```
      MAX2=EED5(M)
      MIN2=EED5(M)
      DO 4000 N=M,1000
      IF (EED5(N) .GT. MAX2) GO TO 4001
      IF (EED5(N) .GE. MIN2) GO TO 4000
      MIN2=EED5(N)
      GO TO 4000
4001  MAX2=EED5(N)
4000  CONTINUE
3000  MAX=MAX1
      MIN=MIN1
      P=1
      L=M
C     TO FIND THE DC VALUE OF THE SINEWAVE
400   NORL=(MAX-MIN)/2
      DC=MAX-NORL
      WRITE (6,15) MAX,MIN,NORL
15    FORMAT(' ','MAX=',I10,'MIN=',I10,'NORL=',F10.5)
C     TO FIND THE ROCKET AZIMUTH ANGLE
      DO 100 J=P,L,1
      IF (TI(J).EQ.0) GO TO 109
      I=TI(J)
      VAL=(EED5(I)-DC)/NORL
      IF (VAL .LT. 0.) GO TO 101
     IF (VAL.LT.1) GO TO 50
     VAL=1
   50 ANGLE(I)=DARSIN(VAL)
      A=EED5(I)-EED5(I+1)
      IF (A .NE. 0.) GO TO 102
      A=EED5(I-1)-EED5(I)
102   CONTINUE
      IF (A .LE. 0) GO TO 110
      ANGLE(I)=3.1416-ANGLE(I)
      GO TO 110
101   VAL= O-VAL
     IF (VAL.LT.1) GO TO 2021
     VAL=1
 2021 ANGLE(I)=DARSIN(VAL)
      A=EED5(I-1)-EED5(I)
      IF (A .NE. 0.) GO TO 103
      A=EED5(I)-EED5(I+1)
103   CONTINUE
      IF (A .LE. 0) GO TO 104
      ANGLE(I)=3.1416+ANGLE(I)
      GO TO 110
104   ANGLE(I)=6.2832-ANGLE(I)
110   WRITE(6,105) I,ANGLE(I)
105   FORMAT(' ',I3,F10.5)
109   CONTINUE
100   CONTINUE
      IF (L .EQ. 1000) GO TO 200
      P=M
      L=1000
      MAX=MAX2
```

```
      MIN=MIN2
      GO TO 400
200   CONTINUE
      RETURN
      END
      SUBROUTINE NICEPR(IEED,JARMAX,XICMIN,XIDELC)
      DIMENSION XLINE(20)
      INTEGER*2 IEED(5000)
      DO 10 J=1,JARMAX
      DO 20 I=1,20
20    XLINE(I) = (IEED((J-1)*20+I)-XICMIN)/XIDELC
      IND = (J-1)*20+1
      IF(MOD(IND-1,100) .EQ. 0)WRITE(6,100)
100   FORMAT(1X)
      WRITE(6,200)IND,XLINE
200   FORMAT(1X,I4,1X,20F6.2)
10    CONTINUE
      RETURN
      END
//GO.FT12F001 DD UNIT=TAPE,VOL=SER=AER018,DISP=OLD,
// DCB=(RECFM=U,BLKSIZE=4016),LABEL=(1,BLP)
//GO.FT11F001 DD SYSOUT=A,DCB=(RECFM=FA,LRECL=133)
//GO.SYSIN DD *
 &PARMS NUMHED=4,NUMCAL=5,ICHECK=1,&END
 &TIMES FSTSEC=40,LSTSEC=350,&END
 &CHANS EEDCH1=1,EEDCH2=2,EEDCH3=3,EEDCH4=4,EEDCH5=5,&END
1652:00     LAUNCH TIME (GMT)
       0007 -- EPS ECLLIPSE
  3600   350
/*
```

APPENDIX XIV. Computing Services utilities for examining tapes.

Two examples, one IBM and one Cyber, of running system utilities TPSNIF and EXAMINE are listed below.

The IBM utility TPSNIF lists the lengths of all records, in 16-bit words, on the tape. Record count within files is also performed. This utility is best for examining binary tape structure.

The Cyber utility EXAMINE is designed to examine tape contents. Besides listing record and file lengths, the Cyber EXAMINE utility attempts to interpret the contents of each file. This utility works best with multi-file tapes containing character data.

```
//MIKE1 JOB
/*ID PS=2714,NAME=MCINERNEY
/*ID CODE=
/*ID PRINT=LOCAL
/*ID BIN=49
/*ID EJECT=YES
/*ID IOREQ=9999,LINES=9999,TIME=1
/*SETUP UNIT=TAPE,R=TEMP,ID=(DA029T,NORING)
// EXEC TPSNIF
      CALL TPSNIF
      STOP
      END
//GO.TAPE DD UNIT=TAPE,VOL=SER=DA029T,
// LABEL=(1,BLP),DISP=OLD
/*
```

```
/JOB
MKM.
SIGNON(3KEMVUJ)
BILL,ELEC,PS2714.
GET,OPTION.
PRINT.
GRAB,EXAMINE.
LABEL(TAPE,VSN=AERO21-F193,F=F,LB=KU,FC=6000,NS=1,NT,PO=UR)
EXAMINE,TAPE,N,D=1600.
UNLOAD(TAPE)
```

APPENDIX XV. Computing Services utilities for copying tapes.

Three examples, two IBM and one Cyber, of running system utilities COPY and DEBBY are listed below.

The IBM utility COPY is preferred for duplicating tapes containing parity errors. The method of processing records with parity errors can be selected. The ERROPT=ACC assignment on line eleven of the IBM COPY JCL sets the error handling option to ACCept. With this setting an erroneous record will be copied as misread.

The error handling options are:

ACC - accept the erroneous block
SKP - skip the erroneous block
ABE - abnormally terminate the task.

For diagnostic purposes when an erroneous record is encountered, the number of the record is printed.

Care must be taken when using the ACCept or SKiP options. When using the ACCept option a record is written with a new parity value, so there will be no indication of a bad record when reading the duplicate tape. When using the SKiP option a "time gap" will occur on the duplicate tape because a bad record was skipped. Therefore the error diagnostics must be consulted to prevent the processing of bad records or the propagation of timing errors when using the duplicate tape.

The IBM utility DEBBY and the Cyber utility COPY operate like the IBM COPY utility with EROPT=ABE.

It is recommended that an IBM tape either be converted to a Cyber format tape or be copied directly to another tape: the new tape to be used in subsequent processing. This preserves the integrity of the original tape, so if the new tape becomes unusable the original can be reduplicated.

The Cyber tape drives are less tolerant of degraded tapes than the IBM tape drives. Older IBM (and NASA) tapes cannot be read by the Cyber tape drives. However, the IBM tape drives may still be able to read the tape, so a duplicate tape can be made.

```
//MIKE1 JOB
/*ID PS=2714,NAME='MCINERNEY'
/*ID CODE=
/*ID BIN=49
/*ID EJECT=YES
/*ID IOREQ=100000,LINES=500,TIME=(6,00),REGION=100K
/*SETUP UNIT=TAPE,R=TEM                    )
/*SETUP UNIT=TAPE,R=F336,ID=(MKM003,RING)
// EXEC COPY
//SYSUT1 DD UNIT=TAPE,VOL=SER=(WI21DA),DISP=OLD,
// DCB=(RECFM=U,BLKSIZE=4016,EROPT=ACC),LABEL=(1,BLP,,IN)
//SYSUT2 DD UNIT=TAPE,VOL=SER=(MKM003),DISP=NEW,
// DCB=(RECFM=U,BLKSIZE=4016),LABEL=(1,NL,,IN)
//SYSIN DD *
/*
```

```
//MIKE1 JOB
/*ID PS=2714,NAME='MCINERNEY'
/*ID CODE=
/*ID BIN=49
/*ID EJECT=YES
/*ID IOREQ=100000,LINES=500,TIME=(6,00),REGION=100K
/*SETUP UNIT=TAPE,R=TEMP,ID=(WI20DA,NORING)
/*SETUP UNIT=TAPE8,R=F336,ID=(MKM003,RING)
// EXEC DEBBY,DEN=2,TAPE=WI20DA,DEN8=2,TAPE8=MKM003
//SYSIN DD *
TT TAPE8,TAPE
/*


/JOB
/NOSEQ
MKM.
SIGNON(3KEMVUJ)
BILL,ELEC,PS2714.
GET,OPTION/UN=3KEMVUJ.
PRINT.
RFL,177000.
SETTL,80.
RESOURC(HD=1,PE=1)
LABEL(TP,VSN=WI22FR-TEMP,NT,PO=UR,F=L,LB=KU,D=HD)
LABEL(TP2,VSN=MKM003-F336,NT,PO=UW,LB=KU,F=L,D=PE)
COPY,TP,TP2.
UNLOAD,TP.
UNLOAD,TP2.
```

APPENDIX XVI. Computing Services documentation of subroutine GBYTES.

```
************************************************************************
*               UOILIB  CYBER SUBROUTINE LIBRARY                       *
************************************************************************
*   COMPUTING SERVICES OFFICE / UNIVERSITY OF ILLINOIS AT URBANA       *
************************************************************************
*----------------------------------------------------------------------*
          IDENT  GBYTES
          ENTRY  GBYTE,GBYTES

*  GBYTE AND GBYTES ARE TWO ROUTINES FOR DATA UNPACKING FROM
*  NCAR. THE CODE BELOW (MODIFIED TO MAKE IT FORTRAN-CALLABLE
*  UNDER NOS) WAS TAKEN FROM AN NCAR TECHNICAL NOTE "TECHNIQUES
*  FOR THE PROCESSING,STORAGE, AND EXCHANGE OF DATA",
*  NCAR-TN/IA-93, DATED JANUARY 1974.
*
*  GBYTE AND GBYTES ARE CALLED AS FOLLOWS:
*
*     CALL GBYTE(SOURCE,DEST,OFFSET,BYTE SIZE)
*
*     CALL GBYTES(SOURCE,DEST,OFFSET,BYTESIZE,SKIP,LOOP)
*
*  THE MEANINGS OF THE PARAMETERS ARE:
*
*     SOURCE   : THE SOURCE WORD OR WORDS FROM WHICH DATA IS TO BE
*                UNPACKED. IN GBYTE THIS IS ONE WORD ONLY, BUT MAY
*                BE AN ARRAY IN GBYTES. THE SOURCE DATA IS VIEWED AS
*                A CONTINUOUS STREAM OF BITS, AS THOUGH ALLL THE 60-
*                BIT WORD(S) WERE STRUNG TOGETHER.
*
*     DEST     : THE DESTINATION WORD OR WORDS INTO WHICH THE UNPACKED
*                DATA IS TO BE STORED. EACH "BYTE" OF DATA WHICH IS UN-
*                PACKED FROM THE SOURCE IS STORED IN A WORD OF THE DEST-
*                INATION, RIGHT-JUSTIFIED WITH ZEROS IN THE REST OF THE
*                WORD.
*
*     OFFSET   : THE NUMBER OF BITS AT THE BEGINNING OF THE SOURCE DATA
*                WHICH ARE TO BE SKIPPED BEFORE PICKING UP THE FIRST
*                "BYTE".
*
*     BYTESIZE: THE NUMBER OF BITS IN THE "BYTE" (OR BYTES) OF INFOR-
*                MATION BEING UNPACKED FROM THE SOURCE DATA. AFTER
*                "OFFSET" BITS, THE ROUTINE TAKES THE NEXT "BYTESIZE"
*                BITS, AND STORES THEM IN THE FIRST WORD OF THE DESTI-
*                NATION, RIGHT-JUSTIFYING THEM AND FILLING THE REST OF
*                THE DESTINATION WORD WITH ZERO BITS.
*
*     SKIP     : THIS APPLIES ONLY TO GBYTES. AFTER THE FIRST "BYTE"
*                HAS BEEN EXTRACTED FROM THE SOURCE DATA, THE ROUTINE
*                IS TO SKIP OVER THE NEXT "SKIP" BITS BEFORE PICKING
*                UP ANOTHER "BYTESIZE" BITS AND STUFFING THEM IN THE
*                SECOND WORD OF THE DESTINATION. THIS GOES ON,
```

```
*               SKIPPING "SKIP" BITS AND PICKING UP "BYTESIZE" BITS,
*               UNTIL "LOOP" BYTES HAVE BEEN EXTRACTED FROM THE SOURCE.
*
*     LOOP    : THIS APPLIES ONLY TO GBYTES. IT IS THE TOTAL NUMBER OF
*               "BYTES" OF DATA BEING EXTRACTED FROM THE SOURCE.
*
* ** RESTRICTIONS **
*
*           IN THE PRESENT VERSION OF THE PROGRAM FOR THE CYBER,
*           "OFFSET" MUST BE LESS THAN 60. ALSO, BYTESIZE MAY NOT
*           EXCEED 60. SKIP MAY BE ARBITRARILY LARGE.
*
*           THE ROUTINE MAKES NO CHECKS FOR "FUNNY" VALUES OF
*           OFFSET, BYTESIZE, SKIP, OR LOOP. YOU MUST MAKE SURE THEY
*           ARE OKAY WHEN YOU CALL THE ROUTINE.
*
*  JUNE 1977 AT THE UNIVERSITY OF ILLINOIS AT URBANA
*
* END WRITEUP
```

REFERENCES

Accardo, C. A., L. G. Smith and G. A. Pintal (1972), Rocket observations of solar x-rays during the eclipse of 7 March 1970, J. Atmos. Terr. Phys., 34, 613-620.

Aikin, A. C. (1962), The sunrise absorption effect observed at low frequencies, J. Atmos. Terr. Phys., 23, 287-300.

Aikin, A. C., J. A. Kane and J. Troim (1964), Some results of rocket experiments in the quiet D region, J. Geophys. Res., 69, 4621-4628.

Appleton, E. V. (1932), Wireless studies of the ionosphere, J. Inst. Elec. Eng., 71, 642-650.

Baldwin, R. B. (1965), A Fundamental Survey of the Moon, McGraw Hill, New York.

Bliss, H. M. and L. G. Smith (1980), Rocket observations of solar radiation during the eclipse of 26 February 1979, Aeron. Rep. No. 93, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Bowhill, S. A. (1970), Ionospheric effects in solar eclipses, in: Solar Eclipses and the Ionosphere, ed. M. Anastassiades, Plenum Press, New York-London.

Brace, L. H., H. G. Mayr, M. W. Pharo III, L. R. Scott, N. W. Spencer and G. R. Carigan (1972), The electron heating rate and ion chemistry in the thermosphere above Wallops Island during the solar eclipse of 7 March 1970, J. Atmos. Terr. Phys., 34, 673-688.

Brekke, A. (1975), The effective recombination coefficient measured in the auroral E-region during a sudden commencement electron precipitation event, J. Atmos. Terr. Phys., 37, 825-833.

Cain, J. C., S. J. Hendricks, R. A. Langel and W. V. Hudson (1967), A proposed model for the international geomagnetic reference field - 1965,

J. Geomagn. Geoelect., 19, 335-355.

Edwards, B. (ed.) (1975), Research in Aeronomy: April 1 - September 30, 1975, Prog. Rep. 75-2, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Edwards, B. (ed.) (1980), Research in Aeronomy: October 1, 1979 - March 31, 1980, Prog. Rep. 80-1, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Fillinger, R. W., Jr., E. A. Mechtly and E. K. Walton (1976), Analysis of sounding rocket data from Punta Chilca, Peru, Aeron. Rep. No. 73, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Fries, K. L., H. D. Voss and L. G. Smith (1979), A rocket-borne energy spectrometer using multiple solid-state detectors for particle identification, Aeron. Rep. No. 91, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Gilchrist, B. E. and L. G. Smith (1979), Rocket radio measurement of electron density in the nighttime ionosphere, Aeron. Rep. No. 85, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Ginther, J. C. and L. G. Smith (1975), Studies of the differential absorption rocket experiment, Aeron. Rep. No. 64, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Grun, A. E. (1957), Lumineszenz-photometrische messunger der energie absorption in strahlungsfeld von elektronengvellen eindimensionaler fall in Luft, Zeitschrift fur Naturforschung, 124, 89-95.

Hartree, D. R. (1931), The propagation of electromagnetic waves in a refracting medium in a magnetic field, Proc. Cambridge Phil. Soc., 27, 143-162.

Hultqvist, B. (1963), On the height distribution of the ratio of negative ion and electron densities in the lowest ionosphere, J. Atmos. Terr. Phys., 25, 225-240.

Klaus, D. E. and L. G. Smith (1978), Rocket observations of electron-density irregularities in the equatorial ionosphere below 200 km, Aeron. Rep. No. 80, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Mechtly, E. A., P. E. Monro, N. Golshan and R. S. Sastry (1970), Fortran programs for calculating lower ionosphere electron densities and collision frequencies from rocket data, Aeron. Rep. No. 37, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Mechtly, E. A., C. F. Sechrist, Jr. and L. G. Smith (1972), Electron loss coefficients for the D-region of the ionosphere from rocket measurements during the eclipses of March 1970 and November 1966, J. Atmos. Terr. Phys., 34, 641-646.

Meira, L. G. (1971), Rocket measurements of upper atmospheric nitric oxide and their consequences to the lower ionosphere, J. Geophys. Res., 76, 202-212.

Narcisi, R. S., A. D. Bailey, L. E. Wlodyka and C. R. Philbrick (1972), Ion composition measurements in the lower ionosphere during the November 1966 and March 1970 solar eclipses, J. Atmos. Terr. Phys., 34, 647-658

Narcisi, R. S., C. Sherman, L. E. Wlodyka and J. C. Ulwick (1974), Ion composition in an IBC Class II aurora. 1. Measurements. J. Geophys. Res., 79, 2843-2847.

Nicolet, M. and A. C. Aikin (1960), The formation of the D region of the ionosphere, J. Geophys. Res., 65, 1469-1483.

Oliver, W. L. and S. A. Bowhill (1973), Investigation of the ionospheric response to the solar eclipse of 7 March 1970 by the Thomson scatter radar technique at the Millstone Hill ionospheric observatory, Aeron. Rep. No. 51, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Omholt, A. (1971), The Optical Aurora, Springer-Verlag, New York, Heidelberg, Berlin.

Rees, M. H. (1963), Auroral ionization and excitation by incident energetic electrons, Planet. Space Sci., 11, 1209-1218.

Rishbeth, H. and O. K. Garriott (1969), Introduction to Ionospheric Physics, Academic Press, New York and London.

Roth, P. (ed.) (1982), Research in Aeronomy: October 1, 1981 - March 31, 1982, Prog. Rep. 82-1, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Sears, R. D. (1972), Analysis of the 1966 solar eclipse data, DNA2863F, LMSC, D246526, Lockheed Palo Alto Res. Lab., California.

Sechrist, C. F., Jr. (1977), The ionospheric D region, National Research Council, Studies in Geophysics, The Upper Atmosphere and Magnetosphere, National Academy of Sciences, 102-116.

Sen, H. K. and A. A. Wyller (1960), On the generalization of the Appleton-Hartree magnetoionic formulas, J. Geophys. Res., 65, 3931-3950.

Slekys, A. G. and E. A. Mechtly (1970), Aeronomy laboratory system for digital processing of rocket telemetry tapes, Aeron. Rep. No. 35, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Smith, L. G. (1969), Langmuir Probes in the Ionosphere, Small Rocket Instrumentation Techniques, North-Holland Publ., Amsterdam.

Smith, L. G. (1972), Rocket observations of solar UV radiation during the eclipse of 7 March 1970, J. Atmos. Terr. Phys., 34, 601-611.

Smith, L. G., E. K. Walton and E. A. Mechtly (1978), Vertical incidence absorption calculated using electron density profiles from rocket experiments and comparison with observations during the winter anomaly, J. Atmos. Terr. Phys., 40, 1185-1197.

Smith, M. J., D. A. Bryant and T. Edwards (1980), Pulsations in auroral electrons and positive ions, J. Atmos. Terr. Phys., 42, 167-178.

Swider, W. (1972), E-region model parameters, J. Atmos. Terr. Phys., 34, 1615-1626.

Swider, W. and R. S. Narcisi (1977), Auroral E-region: Ion composition and nitric oxide, Planet. Space Sci., 25, 103-116.

Szendrei, M. E. and M. W. McElhinny (1956), Preliminary results of the ionospheric solar eclipse of 25 December 1954, J. Atmos. Terr. Phys., 8, 108-114.

Thomas, L. (1974), Recent developments and outstanding problems in the theory of the D region, Radio Sci., 9, 121-136.

Voss, H. D. and L. G. Smith (1974), Design and calibration of a rocket-borne electron spectrometer for investigation of particle ionization in the nighttime midlatitude E region, Aeron. Rep. No. 62, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Voss, H. D. and L. G. Smith (1977), Energetic particles and ionization in the nighttime middle and low latitude ionosphere, Aeron. Rep. No. 78, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Whitten, R. C. and I. G. Poppoff (1965), Physics of the Lower Ionosphere, Prentice-Hall Inc., Englewood Cliffs, N.J.

Whitten, R. C. and I. G. Poppoff (1971), Fundamentals of Aeronomy, John Wiley & Sons, Inc., New York, London, Sydney, Toronto.

Zimmerman, R. K., Jr. and L. G. Smith (1980), Rocket measurements of electron temperature in the E region, Aeron. Rep. No. 92, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.