2mif

UNIVERSITY OF ILLINOIS
URBANA

AERONOMY REPORT
NO. 55

ANALYSIS OF PARTIAL-REFLECTION DATA FROM THE SOLAR ECLIPSE OF JULY 10, 1972

by
T. A. Bean
S. A. Bowhill

October 1, 1973

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

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

CITATION POLICY

The material contained in this report is preliminary information circulated rapidly in the interest of prompt interchange of scientific information and may be later revised on publication in accepted aeronomic journals. It would therefore be appreciated if persons wishing to cite work contained herein would first contact the authors to ascertain if the relevant material is part of a paper published or in process.

UILU-Eng-73-2505

AERONOMY REPORT

NO. 55

ANALYSIS OF PARTIAL-REFLECTION DATA FROM THE SOLAR ECLIPSE OF JULY 10, 1972

by

T. A. Bean
S. A. Bowhill

October 1, 1973

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

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

ABSTRACT

Partial-reflection data collected for the eclipse of July 10, 1972 as well as for July 9 and 11, 1972, are analyzed to determine eclipse effects on *D*-region electron densities. The partial-reflection experiment was set up to collect data using an on-line PDP-15 computer and DECtape storage. Except for a couple of changes, the experiment was the same setup as used by *Birley and Sechrist* [1971]. The electron-density profiles show good agreement with results from other eclipses. The partial-reflection programs were changed after the eclipse data collection to improve the operation of the partial-reflection system. These changes were mainly due to expanded computer hardware and have simplified the operations of the system considerably.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

A solar eclipse can be thought of as the obscuration of solar radiation by the intervention of the moon between the sun and a point on the earth. This obscuring of the sun is a function of time which varies with the location on the earth, altitude, and the type of radiation. Depending on the wavelength of solar radiation and the ionospheric constituents, solar radiation can cause three chemical processes known as dissociation, ionization and excitation [*Whitten and Poppoff*, 1971]. The variation in solar radiation with time during a solar eclipse is given as an obscuration function and varies according to the different wavelengths of radiation. The obscuration function for visible light is easily calculated, being just that for the visible disk. Figure 1.1 shows this obscuration function for the eclipse of July 10, 1972 at 75 km altitude above the University of Illinois Aeronomy Field Station located near Urbana. At this location the eclipse was partial, with about 60% maximum obscuration. The obscuration functions for various other radiations during a total eclipse are shown in Figure 1.2 [*Sears*, 1972]. Notice the large difference between the obscuration functions for ultraviolet radiation and X-rays.

Solar radiation with wavelengths less than 2900Å causes various chemical reactions in the ionosphere [*Whitten and Poppoff*, 1971] with the most pronounced effects occurring in the *D*-region (50 to 90 km). For example, *Turco and Sechrist* [1970] show two orders of magnitude change in the electron density and more than three orders of magnitude change in CO_3^- and CO_4^- at 75 km during sunrise. Certain solar radiations greatly enhance the concentration of positive ions as well as the electron density so that during the daytime, except for during enhanced particle precipitation [*Lauter and Knuth*, 1967], the main ionization source above 70 km

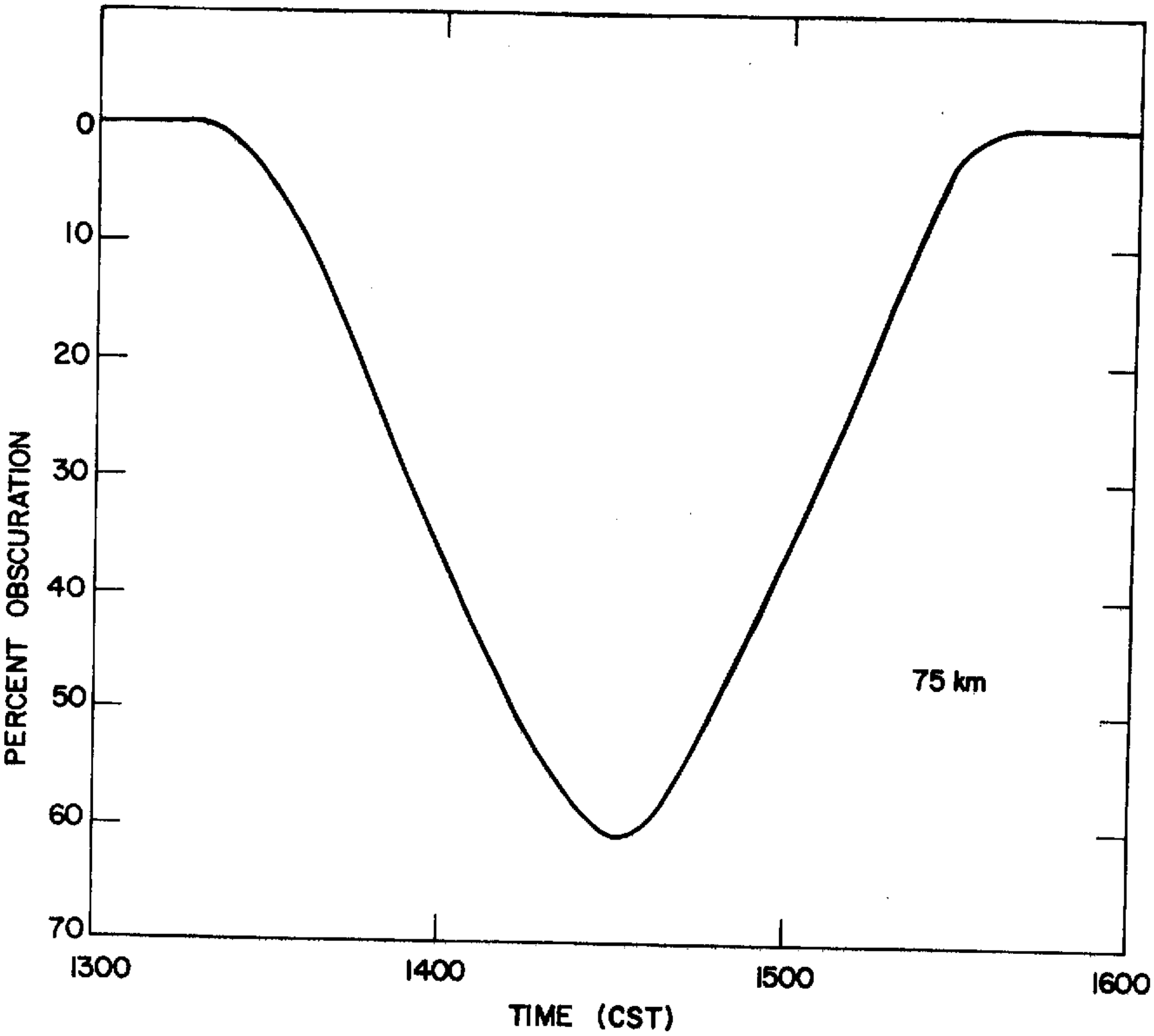

Figure 1.1 The obscuration function of visible light at a height of 75 km for the eclipse of July 10, 1972, near Urbana, Illinois.

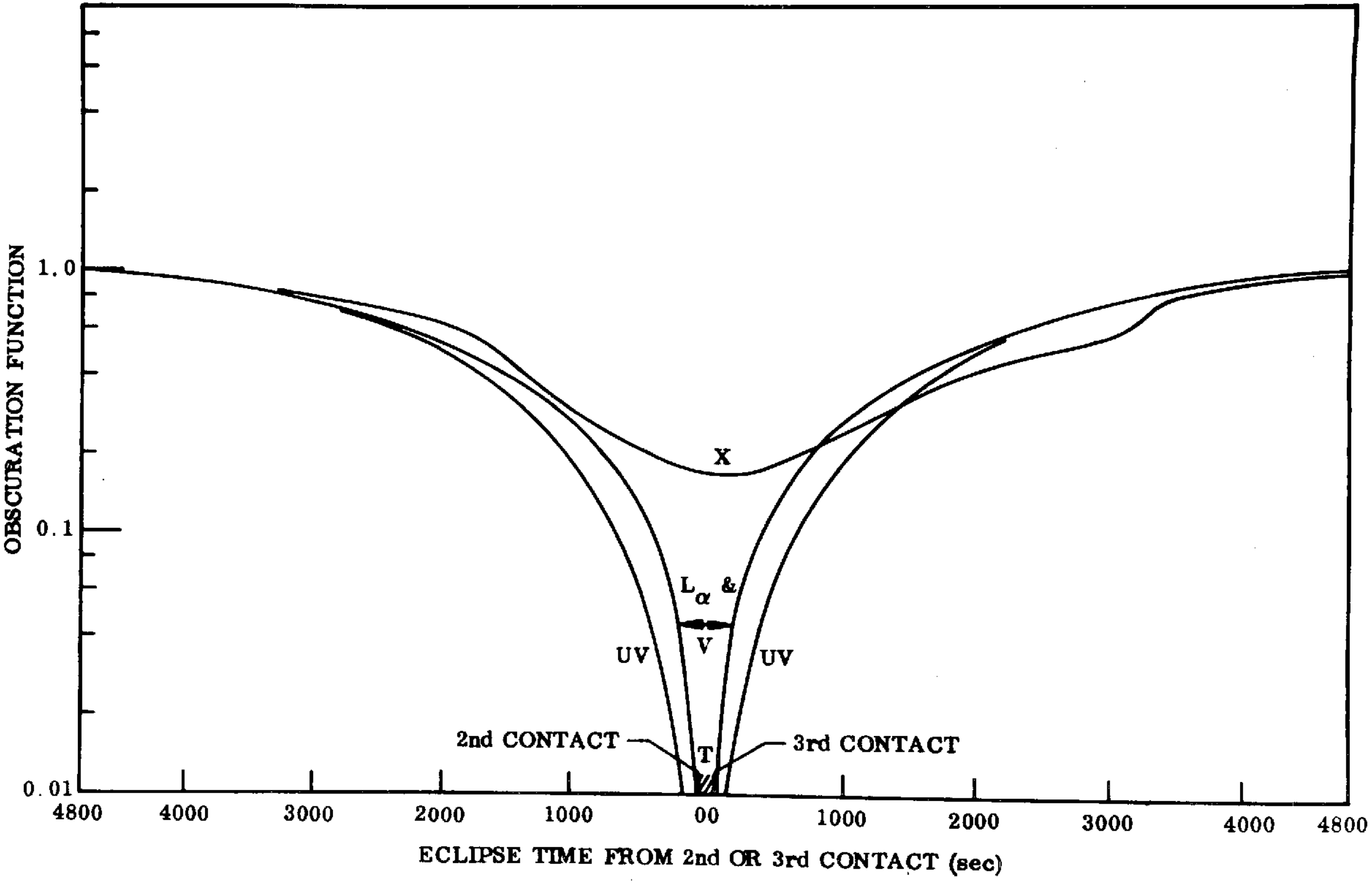

Figure 1.2 Obscuration functions for visible light (V), Lyman alpha (L_α), ultraviolet (UV), and X-ray (X) ionizing fluxes for the 1966 solar eclipse from *Sears* [1972].

is solar radiation as given in Section 2.1. Therefore by correlating the electron densities with the obscuration function for the ionizing radiation in a solar eclipse, values for the production and loss of positive ions and confirmation of the ionizing sources can be obtained.

Data from the *D* region have been obtained by both rocket measurements and ground-based techniques. Although rocket measurements seem to be more accurate [*Mechtly, et al.,* 1967], the amount of data is limited by cost. Ground-based techniques can be set up anywhere and can gather large amounts of data, although the accuracy is not as great, and they are primarily limited to evaluating electron densities. One type of ground-based technique which is discussed in this paper is called the partial-reflection experiment. Data are collected using vertical incident radio waves which are partially reflected from the *D* region. The information obtained can be in one of two forms: differential absorption [*Pirnat and Bowhill,* 1968] and differential phase [*Wiersma and Sechrist,* 1972]. Partial-reflection data using the differential absorption method were collected from 1200 to 1700 CST for the solar eclipse of July 10, 1972, as well as July 9 and 11 as control days. The experiment was set up as described by *Birley and Sechrist* [1971] with two exceptions as described in Chapter 3. The solar and ionospheric conditions for this experiment are given in Chapter 2.

2. PRODUCTION AND LOSS OF THE *D*-REGION IONIZATION

Recently several papers have summarized the knowledge of the *D* region of the ionosphere. *Thomas* [1971] presents an overall review of the *D* region while theoretical models of the *D* region are presented by *Sechrist* [1972], *Ferguson* [1971], *Donahue* [1972], and *Radicella and Stowe* [1970]. The *D* region is perhaps the most complicated part of the ionosphere as well as the most difficult part from which to obtain accurate data. The chemical composition is dependent on height and solar zenith angle [*Thomas,* 1971]; although it consists of neutral constituents, positive ions, negative ions, and free electrons, this chapter is mainly concerned with the processes of formation and loss of free electrons during the daytime (solar zenith angles less than 90°) and during a solar eclipse. Using results obtained from measurements on other eclipses, the expected results from the partial-reflection experiment are given.

2.1 *Ionization Sources*

Although there is general agreement on what ionizes the neutral *D*-region constituents, there is some doubt as to the relative importance of each source. The ionization sources for the daytime *D* region at midlatitudes, as given by *Mitra and Rowe* [1972] and by *Aikin* [1972] are:

1) Lyman-α (1216Å) ionizing nitric oxide (NO)
2) 1-8Å X-rays ionizing all constituents
3) 1027-1118Å ultraviolet radiation ionizing metastable $O_2(^1\Delta_g)$
4) Galactic cosmic rays ionizing all constituents.

Along with these sources precipitating electrons may be considered another source of free electrons, but is of prime importance only in the polar regions, at night, or after a magnetic storm [*Lauter and Knuth,* 1967] and will not be considered in this paper.

The primary ionization source below 70 km is considered to be galactic cosmic rays [*Sechrist,* 1972], although its effect may extend as high as 75 km [*Keneshea,* 1967]. The primary ionization source above 70 km is either (1) or (3) depending upon the nitric oxide distribution adopted. Few measurements of nitric oxide have been made, so most distributions available are from theoretical models. Distributions measured by *Barth* [1966] and *Pearce* [1969] are at least an order of magnitude greater than distributions calculated from theoretical models of the ionosphere [*Mitra,* 1966], but distributions measured by *Meira* [1971] below 85 km are about the same as those calculated by *Shimazaki and Laird* [1970]. Using distribution by *Barth* [1966] for NO, the primary ionization source between 70 and 80 km is Lyman-α ionizing NO, but using nitric oxide distributions given by *Shimazaki and Laird* [1972] and photoionization rates for $O_2(^1\Delta_g)$ given by *Hunten and McElroy* [1968], the main ionization source between 70 and 80 km is 1027-1118 Å UV radiation ionizing $O_2(^1\Delta_g)$ [*Thomas*, 1971]. *Somoyajulu and Avadbanulu* [1972] pointed out that according to measurements by *Huffman* , *et al.* [1971], photoionization of $O_2(^1\Delta_g)$ is important only above 80 km making Lyman-α the main ionization source. Figure 2.1 from *Sechrist* [1972] shows ion-pair production rates for various radiation during solar minimum. In any case the distribution of NO is important to the rate of production of free electrons between 70 and 80 km, and the distribution by *Meira* [1971] is used in this paper.

The variation of ionization sources (1) and (3) with respect to solar activity is small [*Thomas,* [1971], but 2-8 Å X-ray flux can change by several orders of magnitude. Typical X-ray fluxes for different solar activity as given by *Aikin* [1972] are less than 4 x 10^{-3} ergs cm^{-2} sec^{-1} for a quiet sun, between 4 x 10^{-4} and 4 x 10^{-3} ergs cm^{-2} sec^{-1} for moderate sun, and greater than 4 x 10^{-3} ergs cm^{-2} sec^{-1} for an active sun. A solar flare on July 11 at

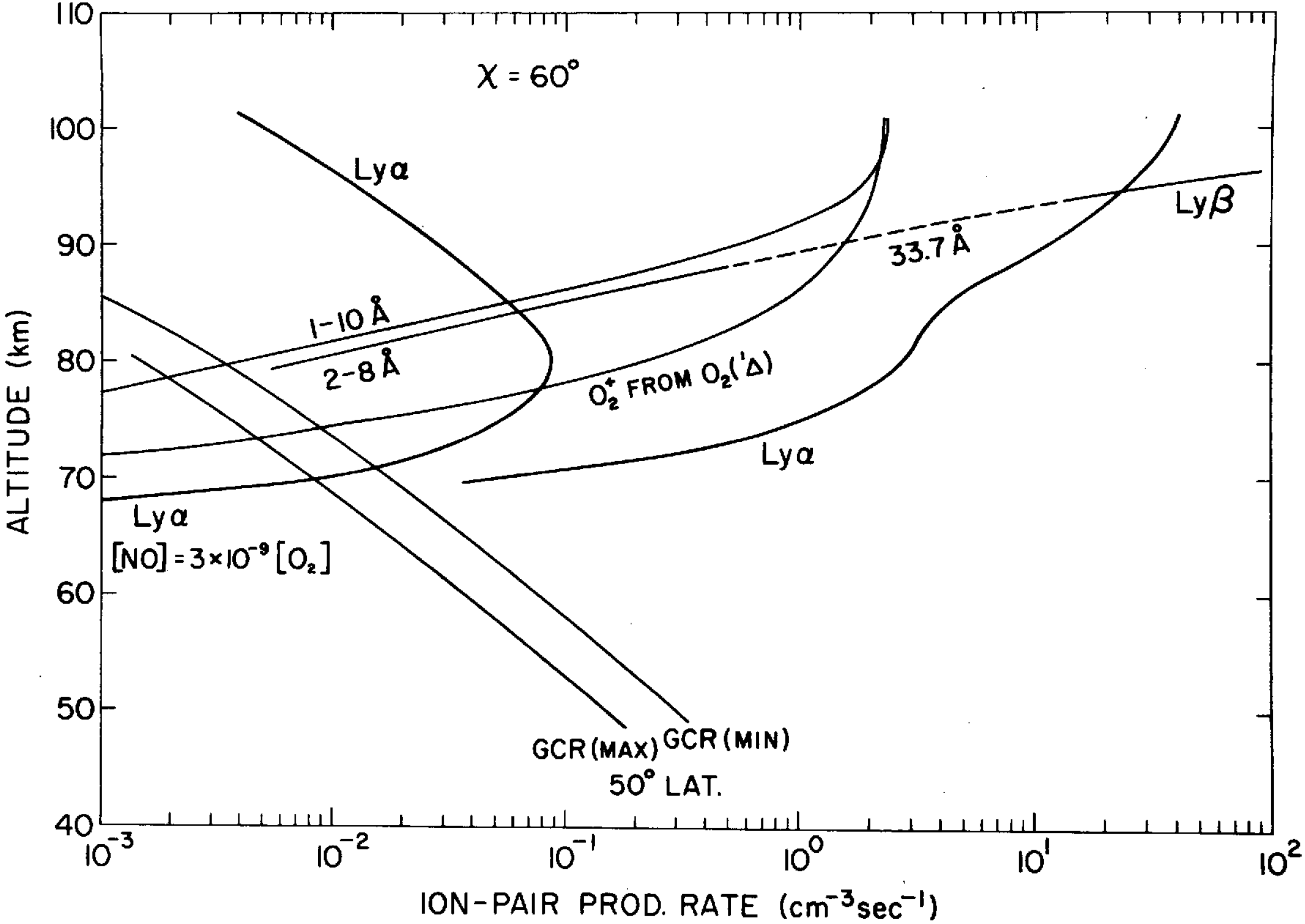

Figure 2.1 Ion-pair production rates from various *D*-region ionization sources as given by *Sechrist* [1972].

8:10 AM CST produced a 2-8 Å X-ray flux of 1.5 x 10^{-2} ergs cm^{-2} sec^{-1}. With an active sun or a solar flare 2-8 Å X-ray ionization can become the primary source of ionization. The 2-8 Å flux for July 9, 10, 11 in Figure 2.2 from the Solar Geophysical Data, 1973 (U. S. Department of Commerce) shows the solar activity to be quiet to moderate. The X-ray flux is expected to have little or no correlation with the electron density of the upper *D* region except for the X-ray burst near 1435 on July 11.

2.2 *Formation of Ions in the* D *Region*

The electron density between 70 and 85 km is dependent on the formation of positive ions. The three main ionization reactions for this region are:

A) $O_2 + h\nu \rightarrow O_2^+ + e$

B) $NO + h\nu \rightarrow NO^+ + e$

C) $N_2 + h\nu \rightarrow N_2^+ + e$

as seen in Figure 2.3 adapted from *Mitra and Rowe* [1972] and *Donahue* [1972], which is a block diagram of the positive-ion chemistry at 75 km. The main loss process for N_2^+ is by the charge-exchange reaction:

D) $N_2^+ + O_2 \rightarrow N_2 + O_2$.

This reaction is very fast (1 x 10^{-10} cm^3 sec^{-1}) [*Fehsenfeld, et al.*, 1965]. Therefore concentrations of N_2^+ are small and the production of O_2^+ is either by photoionization or by charge transfer. Electron production, therefore can be determined by the production of NO^+ and O_2^+ minus the formation of NO^+ by charge exchange reactions shown in Figure 2.3. Since the production of NO^+ is dependent on NO distributions, the production rate of the free electrons also depends on the NO distribution which can differ by at least an order of magnitude (Section 2.1).

The main positive ions between 70 and 80 km are hydrated ions of the form $H^+(H_2O)_n$, n being some number greater than zero [*Narcisi and Bailey*, 1965].

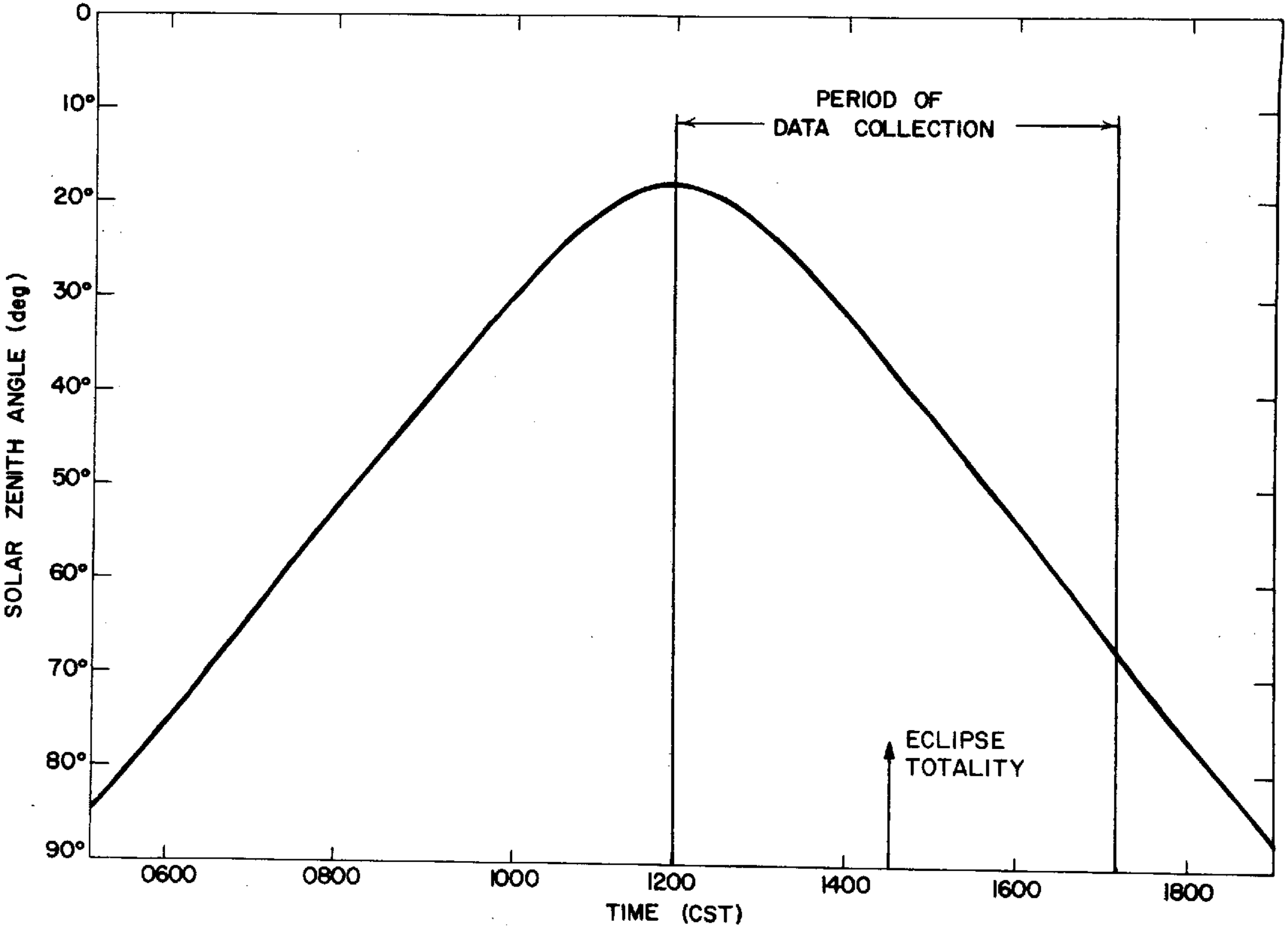

Figure 2.2 Average variations in 2-8 Å X-ray flux during which partial-reflection data were collected on July 9, 10, and 11, 1972.

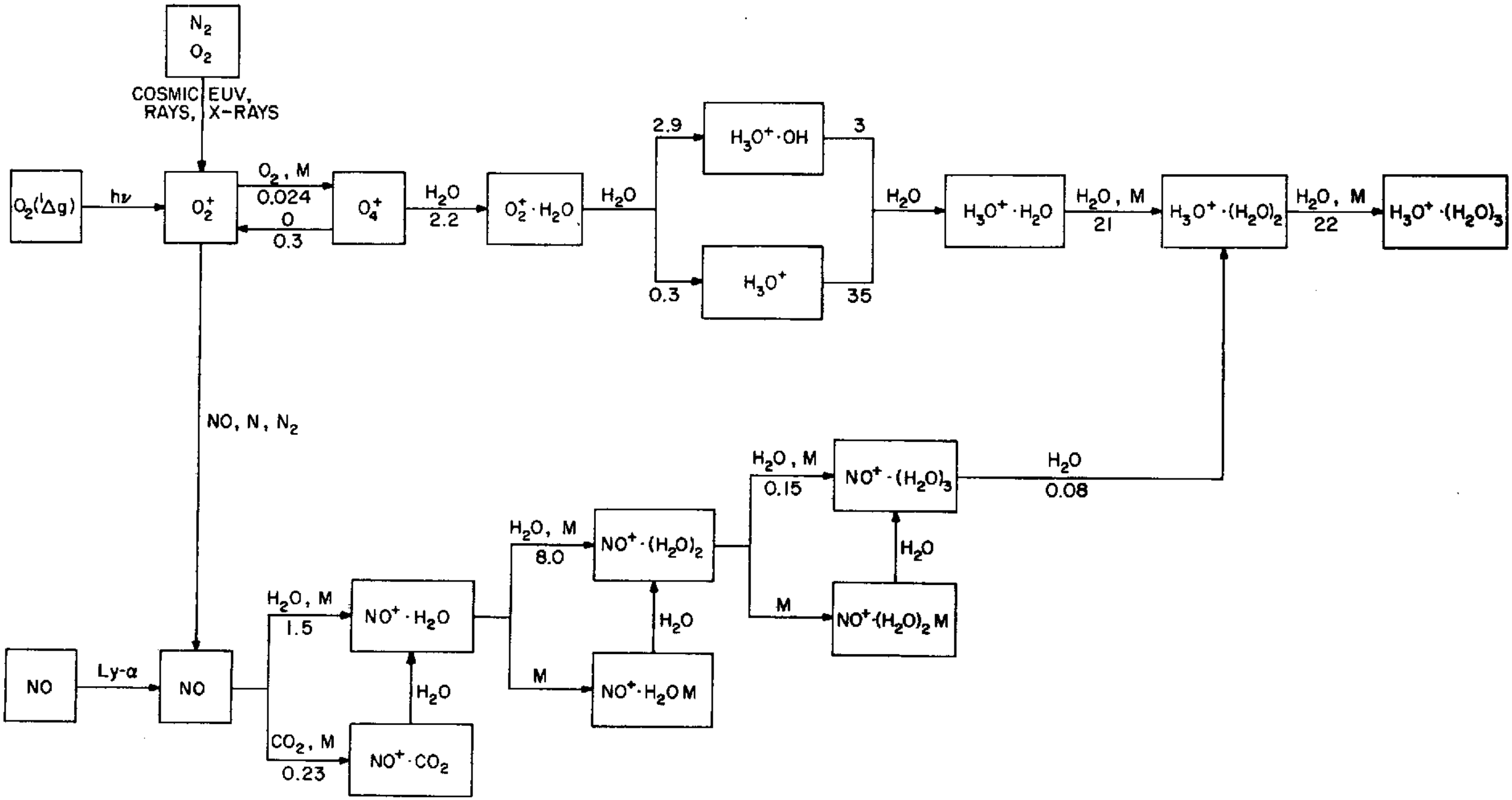

Figure 2.3 Flow diagram of the formation of positive ions including conversion rates [*Donahue,* 1972]. Three-body rate constants are in units of 10^{-28} cm^6 sec^{-1}; two-body rate constants are in units of 10^{-9} cm^3 sec^{-1}. Rate constants not given by Donahue are from *Good, et al.* [1970].

Two basic reaction schemes for the formation of water cluster ions as presented by *Fehsenfeld and Ferguson* [1969] are from NO^+ and beginning with the reaction $O_2^+ + O_2 + M \rightarrow O_4^+ + M$ where M is a third body. Both schemes are given in Figure 2.3. Each scheme raised several questions which are dealt with by *Donahue* [1972]. According to Figure 2.3, NO^+ creates hydrates with masses of 55 and higher, yet 19^+ and 37^+ are the dominant hydrates detected. Also the first three reactions with NO^+ are too slow relative to the loss rate. Problems with the O_2^+ scheme are: it seems to ignore the large NO^+ concentration and the ionization of $O_2(^1\Delta_g)$ seems to be an overestimation according to *Huffman, et al.* [1971], but this may be the main source of water clusters between 77 and 85 km [*Donahue*, 1972]. Even with the large number of hydrated ions, the rapid recombination rate competes with the formation of hydrated ions [*Thomas*, 1971]. This recombination represents the main loss process for free electrons between 70 and 80 km.

The formation of negative ions would constitute a loss of free electrons by the attachment reaction;

E) $e + O_2 + O_2 \rightarrow O_2^- + O_2$.

Figure 2.4 by *Thomas* [1971], giving a scheme for the daytime negative electrons at 65 km, shows reaction (E) to be fast, but the loss reactions

F) $O_2^- + O \rightarrow O_3 + e$

G) $O_2^- + O_2(^1\Delta_g) \rightarrow 2\ O_2 + e$

are much faster. Although the formation of O_4^- is rapid, there is rapid return to O_2^-. The negative ion chemistry is dependent on atomic oxygen and $O_2(^1\Delta_g)$ concentrations. At night these concentrations decrease so that reaction (E) constitutes an important loss process for free electrons.

At eclipse totality free electron production is reduced to that comparable of nighttime electron production, and the production of atomic oxygen and metastable $O_2(^1\Delta_g)$ are also greatly reduced [*Shimazaki and Laird*, 1972]. By

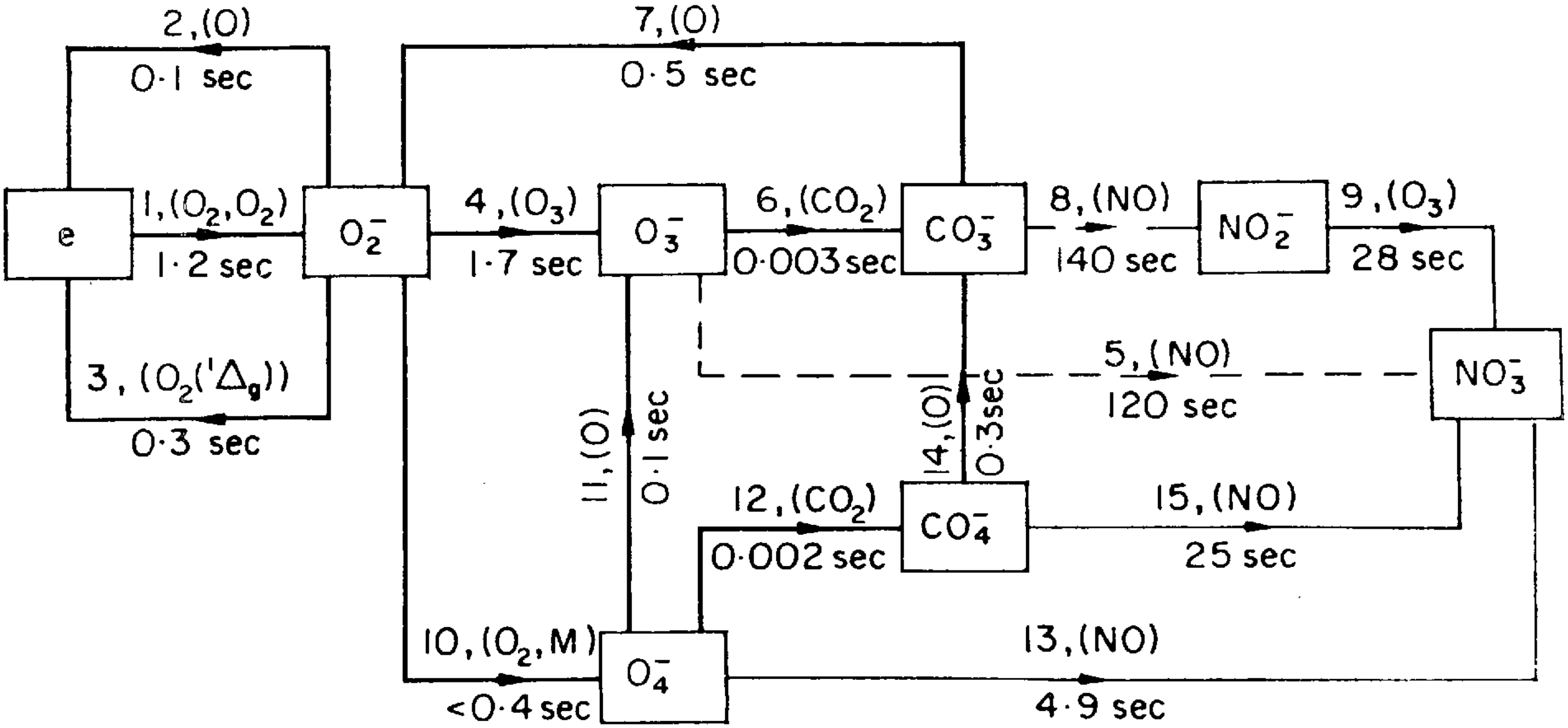

Figure 2.4 Block diagram [*Thomas*, 1971] showing the negative ion chemistry during the day. The lifetimes of electrons and each ion are for a height of 65 km.

comparison of eclipse data, *Mechtly, et al.* [1972] shows the possibility of attachment reactions as being the main loss process at totality. This would mean a large reduction in O and $O_2(^1\Delta_g)$, but the reduction measured by *Hunt* [1965] during an eclipse shows less than an order of magnitude change in atomic oxygen. More measurements of atomic oxygen are needed during eclipses to determine more accurately the loss process for free electrons during totality of a solar eclipse.

2.3 *Recombination*

Above 70 km during the daytime, negative-ion chemistry is not important; so the main loss process of free electrons above 70 km is by recombination with positive ions. The continuity equation for electrons as given by *Whitten and Poppoff* [1971] is:

$$\frac{d[e]}{dt} = \left(\frac{q}{1+\lambda}\right) - (\alpha_D + \lambda\alpha_i)[e]^2 - \left(\frac{[e]}{1+\lambda}\right)\frac{d\lambda}{dt} \tag{2.1}$$

where $[e]$ is the electron density, λ is the ratio of negative ion concentrations to electron densities, q is the ionization rate, α_D is the ion-electron recombination coefficient, and α_i is the ion-ion recombination coefficient.

With the assumption that variation in λ is insignificant, then $d\lambda/dt = 0$ and defining an effective recombination coefficient as $\alpha_{\mathrm{eff}} = \alpha_D + \lambda\alpha_i$, Equation (2.1) reduced to:

$$\frac{d[e]}{dt} = \left(\frac{q}{1+\lambda}\right) - \alpha_{\mathrm{eff}}[e]^2 \tag{2.2}$$

During a solar eclipse at totality, the electron production decreases by several orders of magnitude. Using an ionization rate of zero ($q = 0$), α_{eff} can be obtained from Equation (2.3) for short intervals of time.

$$\alpha_{\text{eff}} = \frac{\Delta[e]}{\Delta t}[e]^2 \tag{2.3}$$

With small changes in the electron density α_{eff} can be obtained by the approximation [*Mitra and Rowe*, 1972]

$$\alpha_{\text{eff}} = q/[e]^2(1 + \lambda) \tag{2.4}$$

Below 70 km the problem is complicated by the presence of negative ions [*Mitra and Rowe*, 1972] for which a time dependent analysis of the negative reaction scheme has to be used [*Thomas*, 1971]. As discussed in Section 2.2, there is the possibility of loss by attachment. Many problems about the loss process still remain unsolved including the question of the NO distribution.

2.4 *Expected Results*

Figure 1.2 by *Sears* [1972] gives the obscuration function for different *D*-region solar ionization sources from the eclipse of 1966. Lyman-α and visible light have the same obscuration function but not so with UV and X-rays. The obscuration function for visible light at Urbana, Illinois for July 10, 1972 (Figure 1.1) is therefore expected to be different from the obscuration function for ultraviolet radiation and X-rays. Using the maps of the sun given in Solar-Geophysical Data, 1972 (U.S. Department of Commerce) and the moon's movement across the sun's disk, an idea of the obscuration function for different solar radiations can be obtained. Since the solar activity during the eclipse was quiet to moderate, the predominate ionization source between 70 and 80 km is expected to be Lyman-α.

The total obscuration is about 60%, therefore data is used from previous eclipses with a similar obscuration and about the same solar zenith angle. The solar zenith angle is shown in Figure 2.5 to be about 37°. Figure 2.6 by *Deeks* [1966] gives various electron densities for an eclipse during March equinox noon at sunspot minimum. Figure 2.7 by *Smith, et al.* [1965] gives electron density distributions for various obscurations of the eclipse of July 20, 1963. In Figure 2.6 the electron density for 60% obscuration shows little change until above 70 km. For Figure 2.7 at 40% obscuration the electron density at 75 km has no change while above and below this altitude show marked changes. Below 75 km the change is, therefore, expected to be no larger than above 75 km and the change is expected to be approximately 36% (from equation (2.4)). Due to the changing solar zenith angle, the magnitude of the slope of the changing electron densities before the maximum obscuration of the sun is expected to be greater than the slope after maximum obscuration.

2.5 *Statement of the Problem*

The purpose of this paper is to present the setting up, collection, and analysis of the partial-reflection data taken before, during and after a solar eclipse and to present changes made in the partial-reflection computer programs in order to simplify the operation and more effectively reject noise.

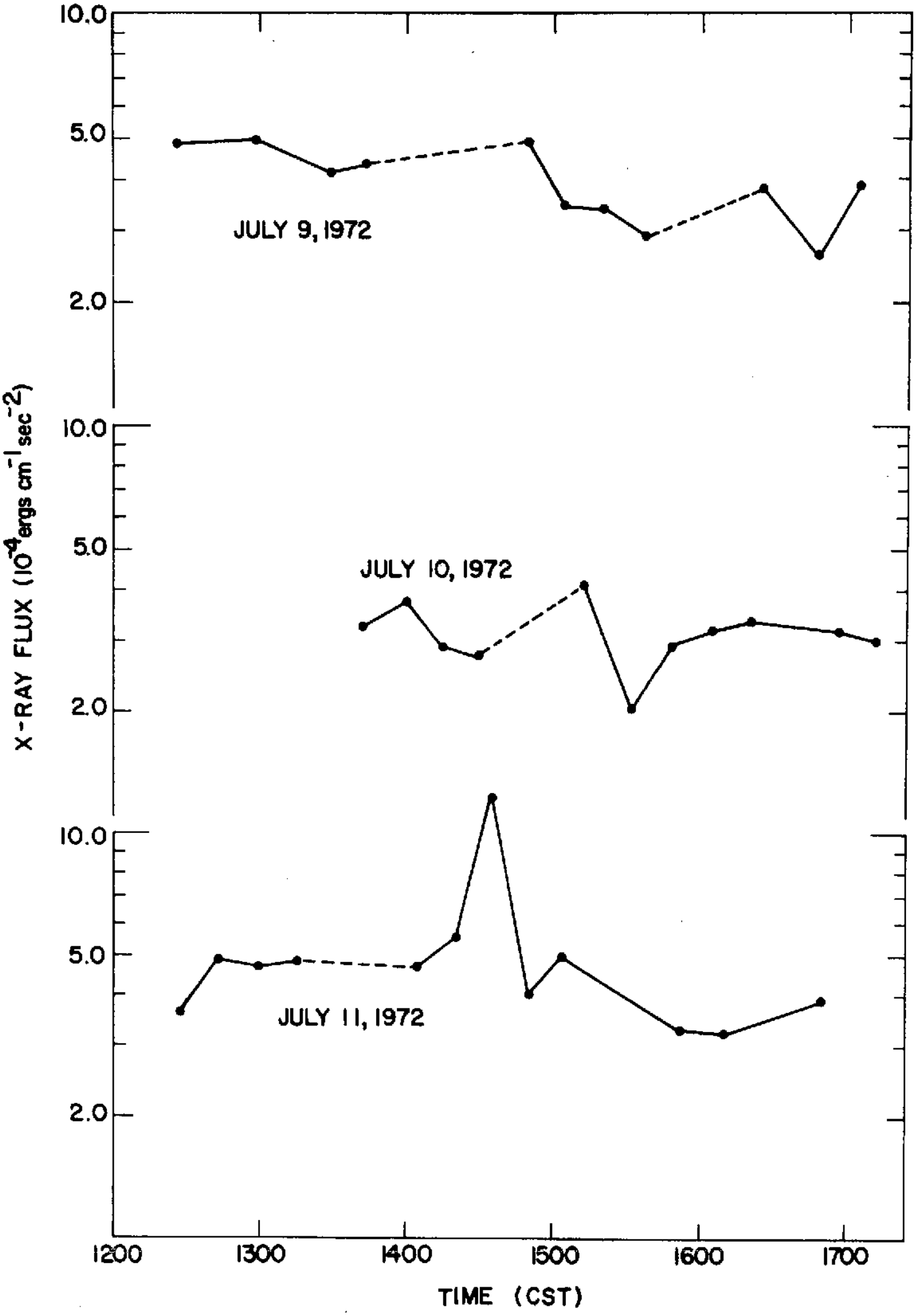

Figure 2.5 The variation of the solar zenith angle for July 10, 1972. The partial-reflection data collected period is shown as well as the time of maximum obscuration for the eclipse.

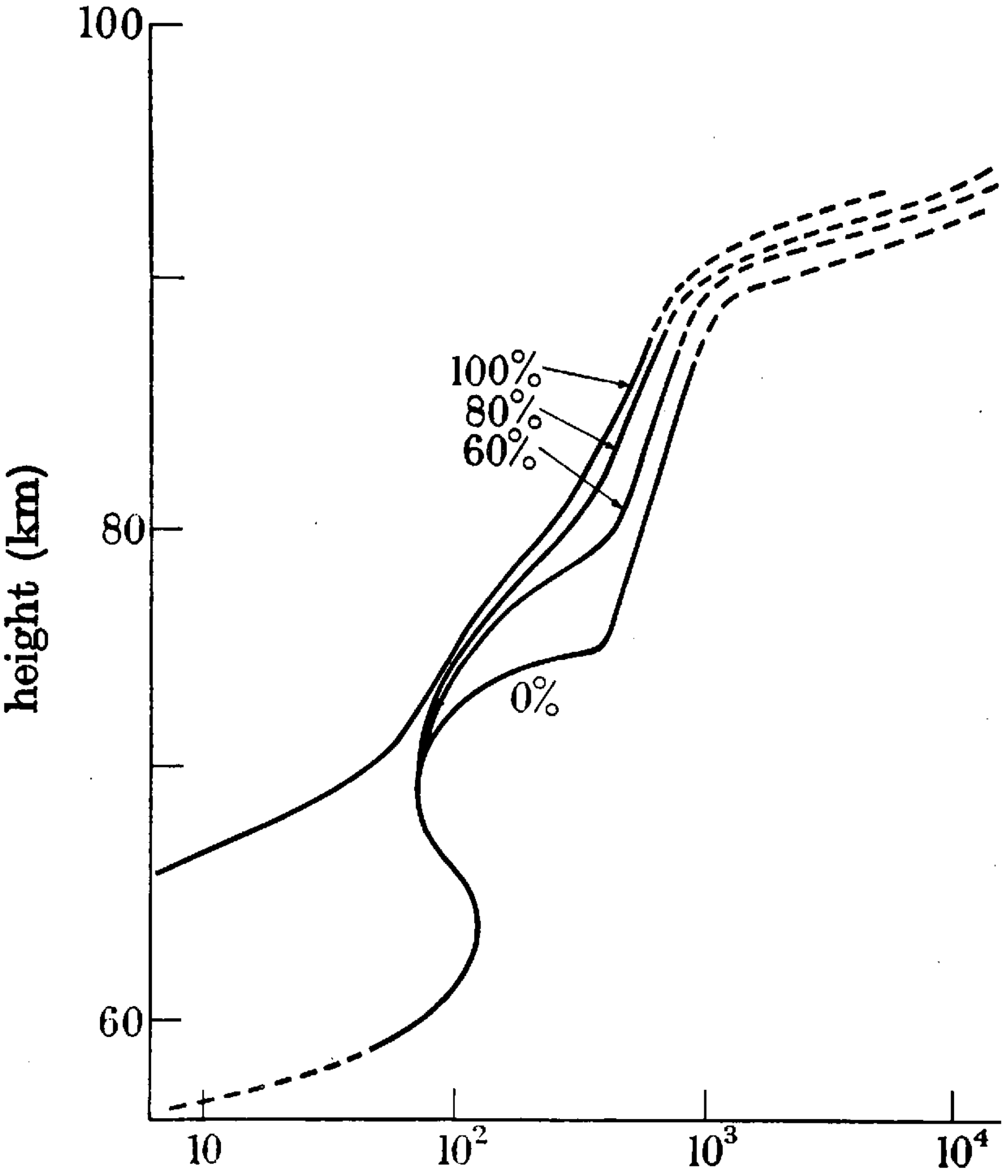

Figure 2.6 Variation of electron density during a solar eclipse at March equinox, mid-day, and sunspot minimum at middle latitudes [*Deeks*, 1966].

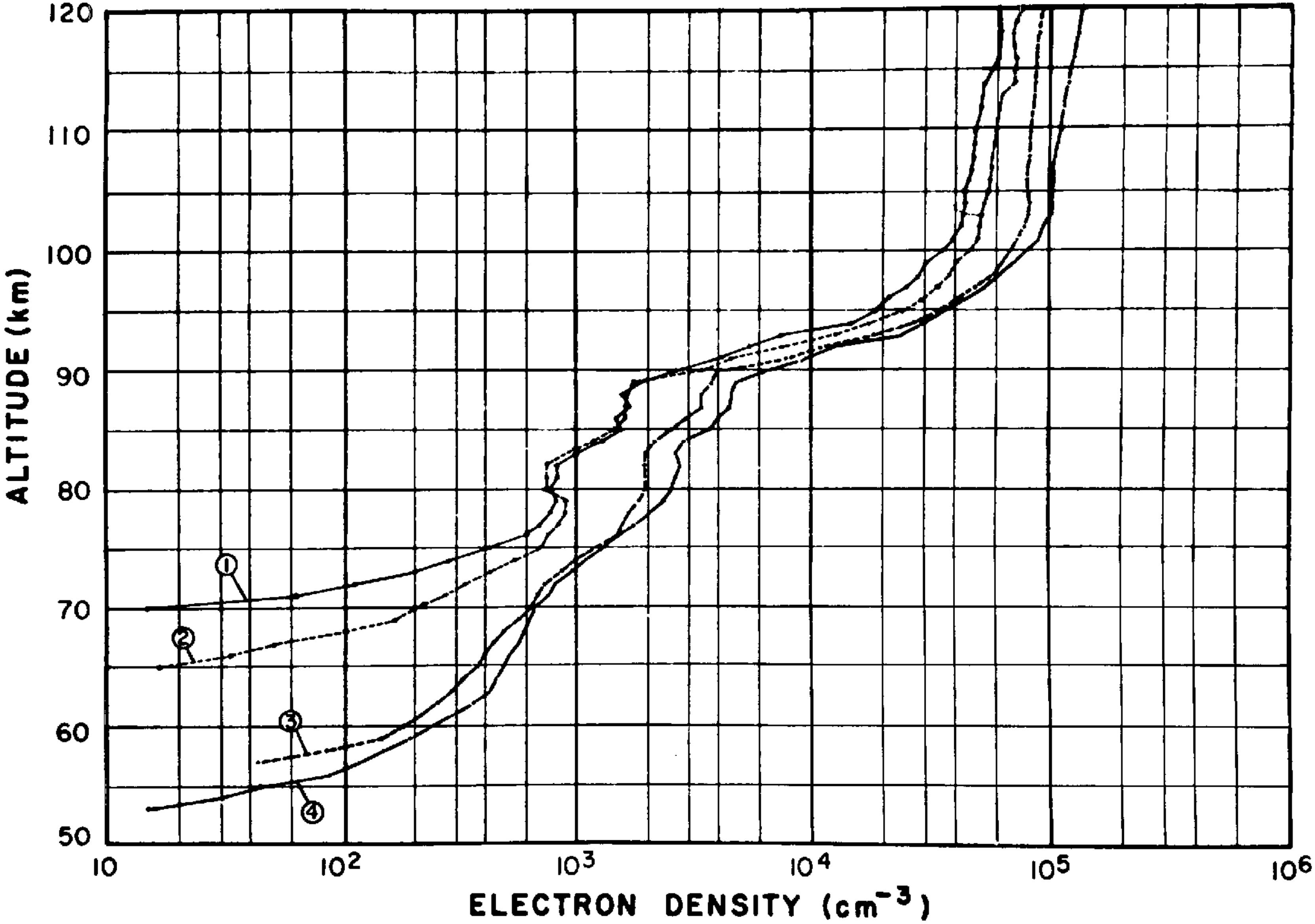

Figure 2.7 Electron-density profiles for the eclipse of July 20, 1963 [*Smith, et al.*, 1965]. Profiles 1, 2, 3, and 4 refer to obscurations of 92%, 86%, 40%, and 2%, respectively. The solar zenith angle was 55° at totality and 61° at 40% obscuration.

3. EXPERIMENTAL TECHNIQUE

The partial-reflection experiment was first performed by *Gardner and Pawsey* [1953]. Electron densities were deduced for 65 to 82 km from partially reflected, circularly polarized radio waves. The transmitter operated at 1 kw during each 30 μsec pulse with a center frequency of 2.28 MHz, and the partially reflected signals were displayed on an A-scan oscilloscope. Several improvements have been made in the experiment and are discussed by *Pirnat and Bowhill* [1968].

Gregory [1956] used an increase in transmitter power of 4 kw and a decrease in the transmitter pulse width to 9 μsec. These changes improved the amplitude and resolution of the partial reflections. *Fejer and Vice* [1959] developed an improved receiving and storing method using a dual-beam cathode-ray tube oscilloscope and camera. The system was operated at 1.83 and 2.63 MHz. *Belrose and Burke* [1964] also operated at two different frequencies (2.66 and 6.275 MHz) and transmitter power of 1 Mw, were able to obtain electron densities from the *D* and *E* region. *Belrose and Burke* [1964] were the first to use the generalized Appleton-Hartree formulas by *Sen and Wyller* [1960] for partial-reflection application.

Using the generalized Appleton-Hartree formulas and several approximations, the ratios of partially reflected extraordinary waves (A_x) to the partially reflected ordinary wave (A_o) for two heights can be used to calculate electron densities [*Pirnat and Bowhill*, 1968 and *Reynolds and Sechrist*, 1970]. The ratio A_x/A_o at each height is inversely related to the absorption by the expression $\exp(2\int_0^h k_x-k_o)$ from which the name differential absorption originates. At the University of Illinois the electron density was calculated directly from these ratios, and as seen in Chapter 4, small changes in these ratios can produce large variations in the electron densities.

Henry [1966] designed and built the hardware for the partial-reflection experiment at the University of Illinois. The transmitter that is presently being used was built for the purpose of making shipboard measurements. This transmitter operates at 40 kw during each 20 μsec pulse and with 5 pulses per second. The center frequency is 2.66 MHz with a 50-ohm unbalanced output. Figure 3.1 shows a block diagram of the transmitter. The reduction of power from the initial 50 kw used is to give longer life to the tubes used, and the pulse is shortened from 50 μsec used by *Henry* [1966] for better height resolution.

Figure 3.2 shows the two antenna arrays used to transmit and receive circularly polarized signals. Each array consists of 30 half-wave dipoles in the north-south direction and 30 in the east-west direction [*Wiersma and Sechrist,* 1972]. Each direction has matching networks that differ by 90° from the other direction of the same array to give a circularly polarized radio wave as shown in Figure 3.3. Each array gives approximately 22 dB gain with the main beam in the vertical direction. The first sidelobe is down 14 dB. Since both arrays are the same, this is a decrease of approximately 30 dB in the sidelobes relative to the main signal which has 44 dB gain. Further details on the antennas are given by *Pirnat and Bowhill* [1968] and *Reynolds and Sechrist* [1970].

3.1 *Development of Receiving and Storing Data*

The receiver, storage and timing controls have had two main changes in the development of the partial-reflection system. The experiment was originally set up using photographic film to store the partially reflected signals as displayed on an oscilloscope (see Figure 3.4). The controlling circuitry or pulser sent pulses of 30 volts to the transmitter, receiver, and camera. The pulser has remained the same with the exception of the addition of extra control circuitry depending on the storage method. The amplitudes of the received signals

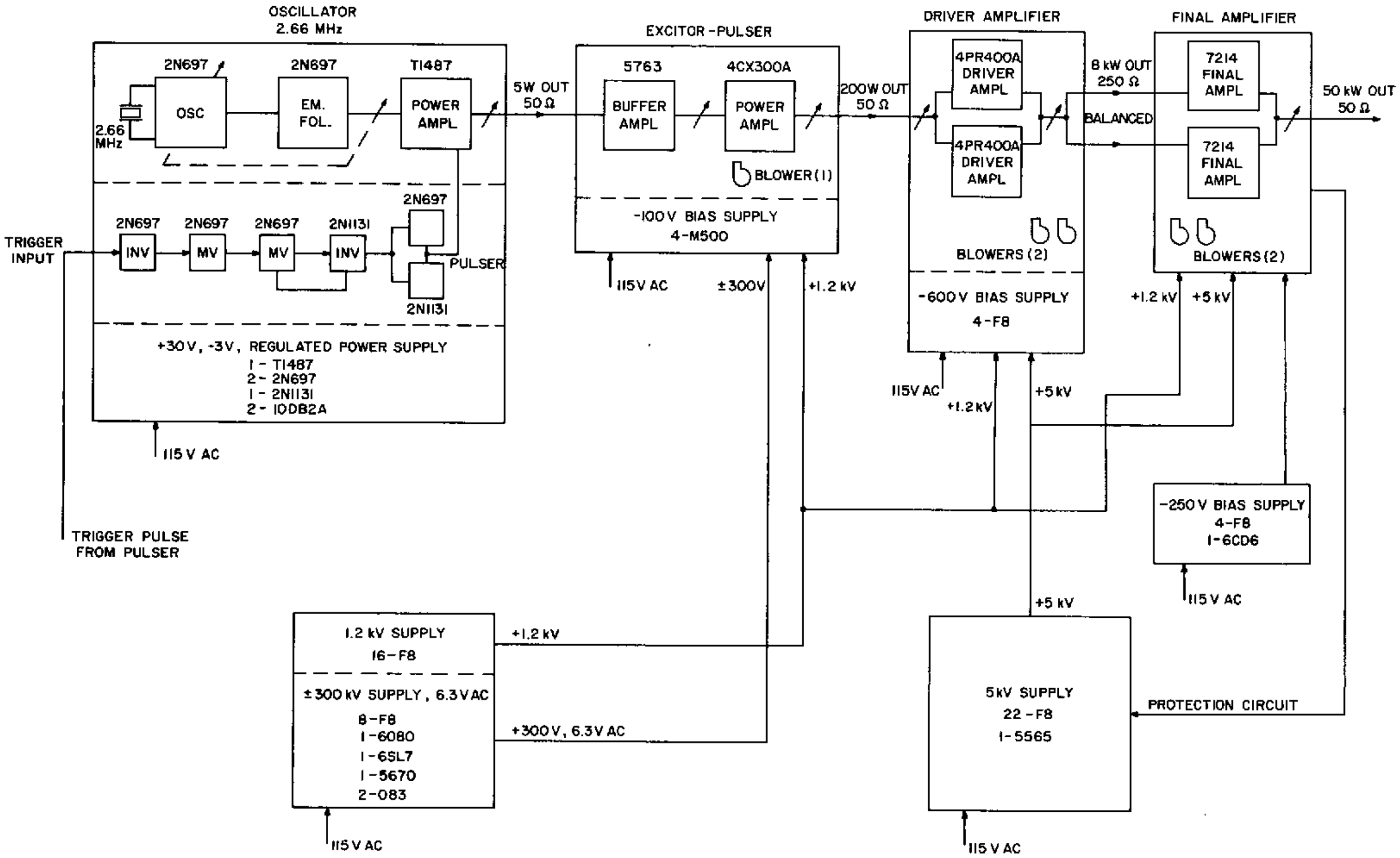

Figure 3.1 Block diagram of the partial-reflection transmitter.

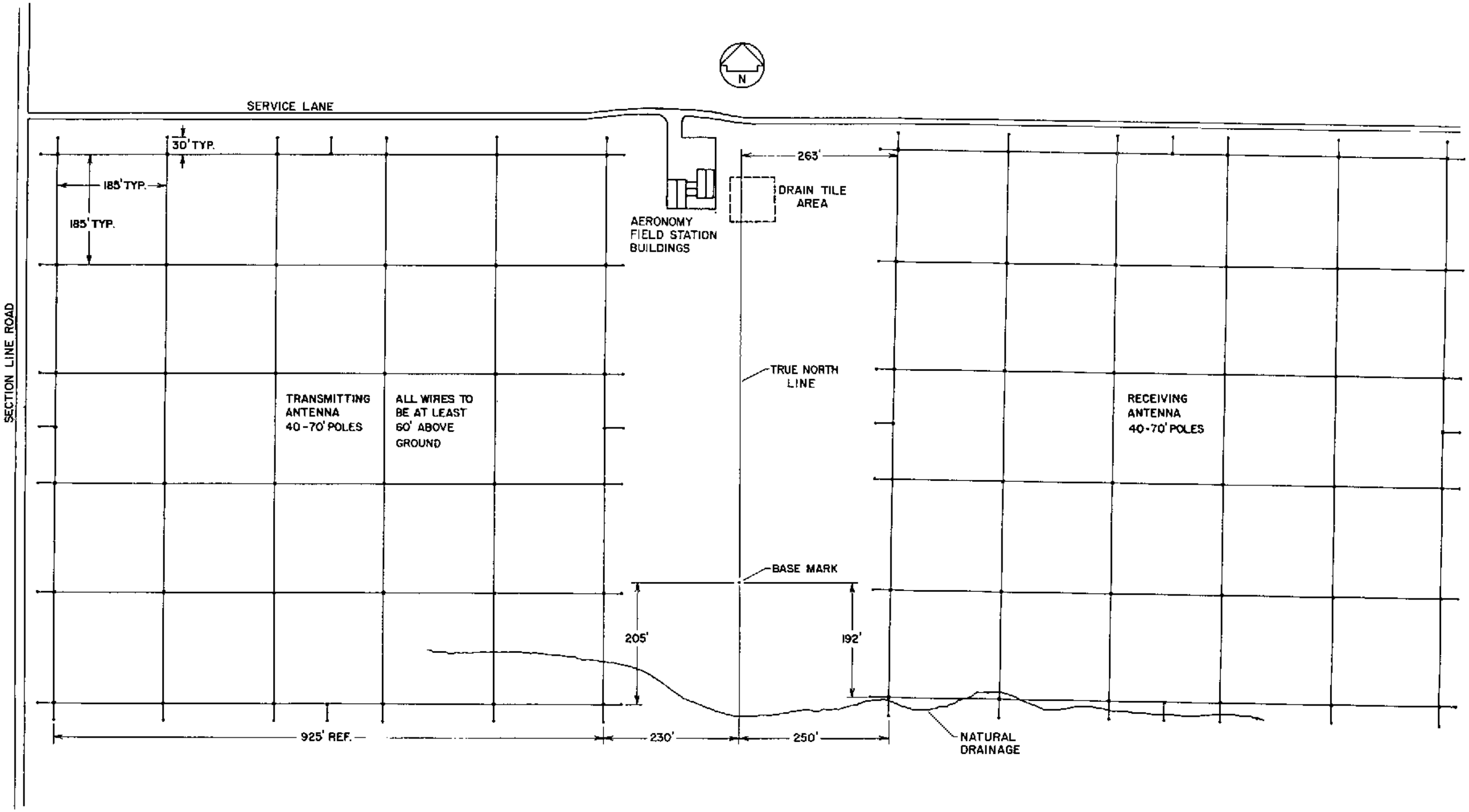

Figure 3.2 Partial-reflection antenna arrays for the Aeronomy Field Station.

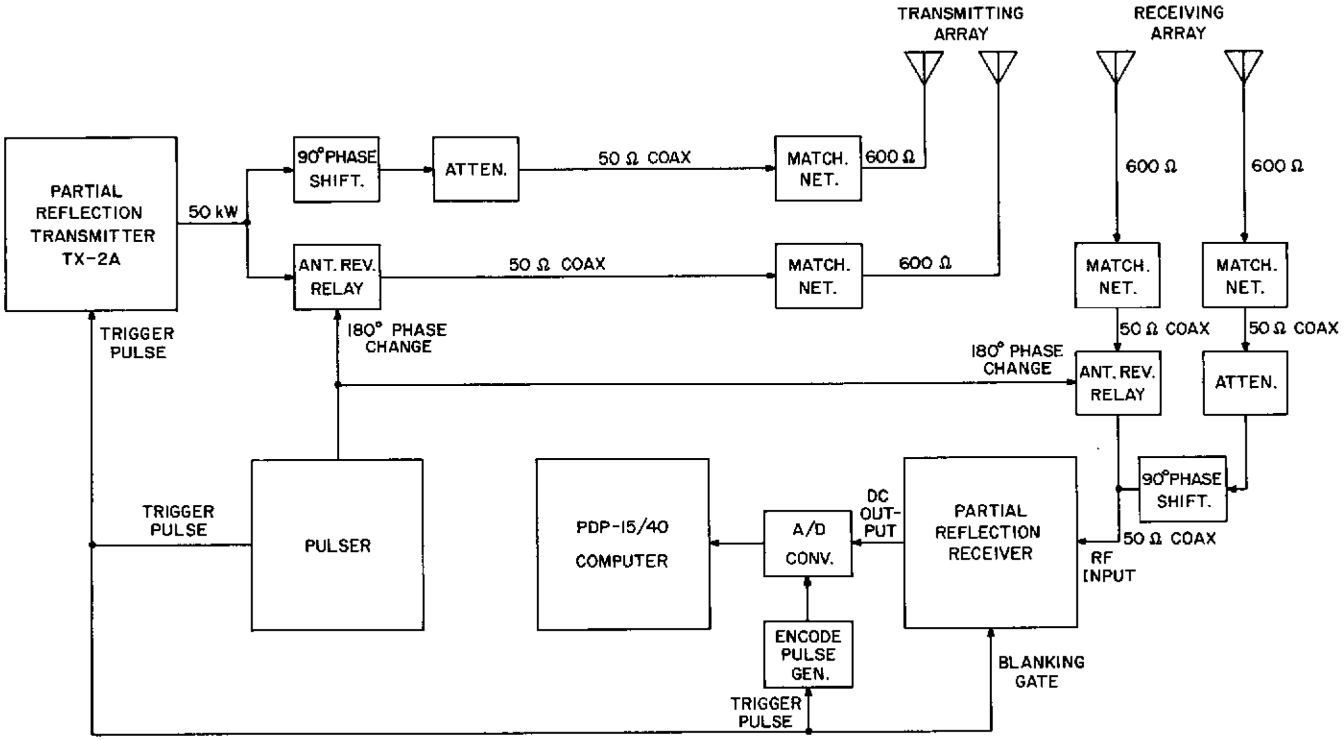

Figure 3.3 Block diagram of the partial-reflection system.

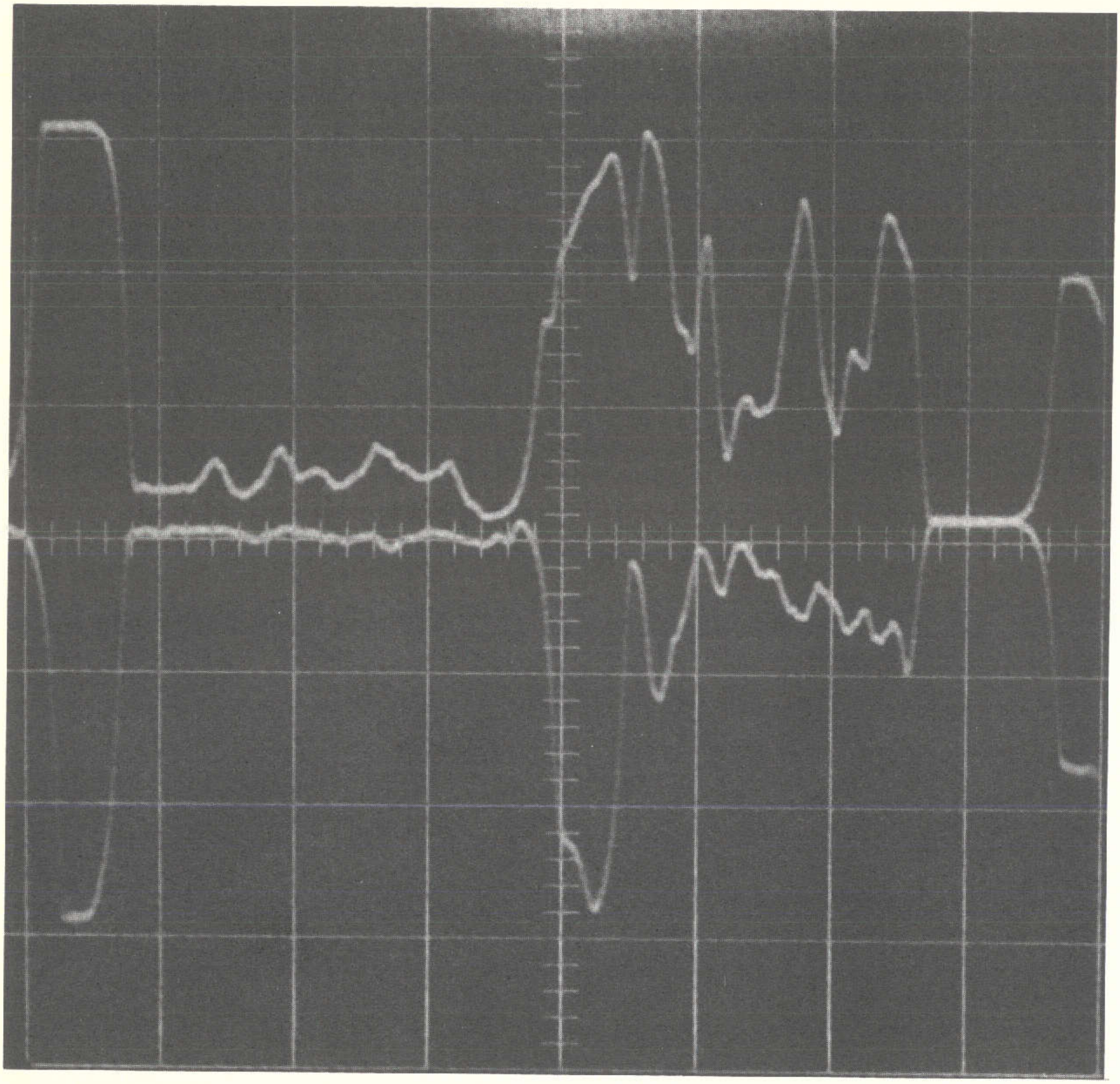

Figure 3.4 Typical frame of data as collected by *Henry* [1966].

were later measured visually and electron densities were obtained. *Pirnat and Bowhill* [1968] shows that there is good correlation between electron densities calculated from the partial-reflection data and from rocket measurements with the transmitter operating at 25 kw of power during a 50 μsec pulse. This system of collection and storage is inexpensive, but the processing of the data to obtain electron densities is very slow and preparation and operation are complicated.

Reynolds and Sechrist [1970] set up data storage on paper tape. Ordinary and extraordinary samples were punched on paper tape for heights corresponding to 75 km and 80 km. Data can be stored at a rate of 30 values of each sample in one minute. From the paper tape the data can then be read into a computer and processed. This data on paper can be used to obtain an electron density for between 75 and 80 km. *Reynolds and Sechrist* [1970] show the results using paper tape compares favorably with results from rocket measurements and with the results published by *Belrose and Burke* [1964]. Although the system has a faster operation than the original system, it produces only one electron density and the added control circuitry is very complex.

Birley and Sechrist [1971] set up the partial-reflection experiment using a PDP-15 computer. The received signals were transmitted to the computer via an analog to digital converter and stored on DECtape to be processed later. The data consisted of four noise samples from 45 to 49.5 km and 21 data samples from 60 to 90 km in 1.5 km increments. The collection rate is 5 sets of 26 samples $\sec^{-1}$. This collection is done alternating between ordinary partial reflection and extraordinary partial reflections. Electron densities obtained by *Birley and Sechrist* [1971] show good agreement with electron densities obtained from rocket measurements between 67.5 and 82.5 km. The other heights suffered from too many rejections due to noise and saturation of the analog to digital converter, small signal to noise ratios, or inaccurate A_x/A_o ratios. Computer storage offers several advantages:

1) A fast rate of data collection (presently limited to the transmitter speed)
2) Data can be stored more compactly and in much larger quantities
3) The controlling circuitry is greatly simplified
4) The data processing is faster
5) $[e]$ can be obtained for every 1.5 km

This type of system also poses several disadvantages:

1) High cost
2) Development of computer software
3) Loss of accuracy in digitizing the data
4) Development of new circuitry and modification of the old for adaption to the A/D converter
5) More complicated operations (operator must know computer operation)

These disadvantages have been reduced with additional equipment and development as given in Section 3.3.

3.2 *Partial-Reflection Data Collection for the Solar Eclipse*

The partial-reflection receiver was interfaced into the PDP-15 computer to obtain data to be processed as described by *Birley and Sechrist* [1971]. Several changes in the receiver and controlling circuitry and the addition of an analog-to-digital converter were required prior to using the computer. A block diagram of the original receiver is shown on page 18 of Aeronomy Report

No. 13, [*Henry*, 1966]. The analog-to-digital converter saturates with an input of one volt or greater and will be damaged with inputs greater than five volts. The maximum output of the receiver was therefore reduced from 10 volts to 1.5 volts by one of the IF amplifiers, and the full-wave bridge diode detector was replaced by a single diode to reduce the nonlinearity of the receiver. A second blanking gate was inserted with the mixer in the RF amplifier module to more completely remove the initial effects of the transmitter pulse. The polarity reversal circuitry was not used but was left intact while the differential amplifier and inverter were replaced by two DC amplifiers on integrated chips.

The block diagram of the modified receiver is shown in Figure 3.5. Figure 3.6 shows the RF module with the extra blanking gate and Figure 3.7 shows the IF amplifier/DC amplifier module with the revisions. Both modules were modificationsof the RF-3 module and IF-6 module respectively, given by *Henry* [1966]. The receiver power supply was unchanged as set up by *Henry* [1966]. Encode pulses as shown in Figure 3.8 were used to control the operation of the A/D converter after *Birley and Sechrist* [1971]. The encode pulse circuitry consists of a 5-volt power supply and 4 monostable multivibrators (Figure 3.9) with a variable timing for length of noise and signal pulses and the delay of each.

Two main modifications were made in the software set up by *Birley and Sechrist* [1971]. For the first change *D. R. Ward* [private communication] set up a computer-controlled synchronization with the external pulser. The timing shown in Figure 3.9 is used to determine which radio wave mode has been received. The computer programs are set up to store only pairs of sets of 26 numbers read from the A/D converter. A set of numbers is read in and assumed to be from a radio wave of ordinary mode. The computer's clock is set for 150 μsec and the computer waits for another set of numbers. If another set is not read in prior

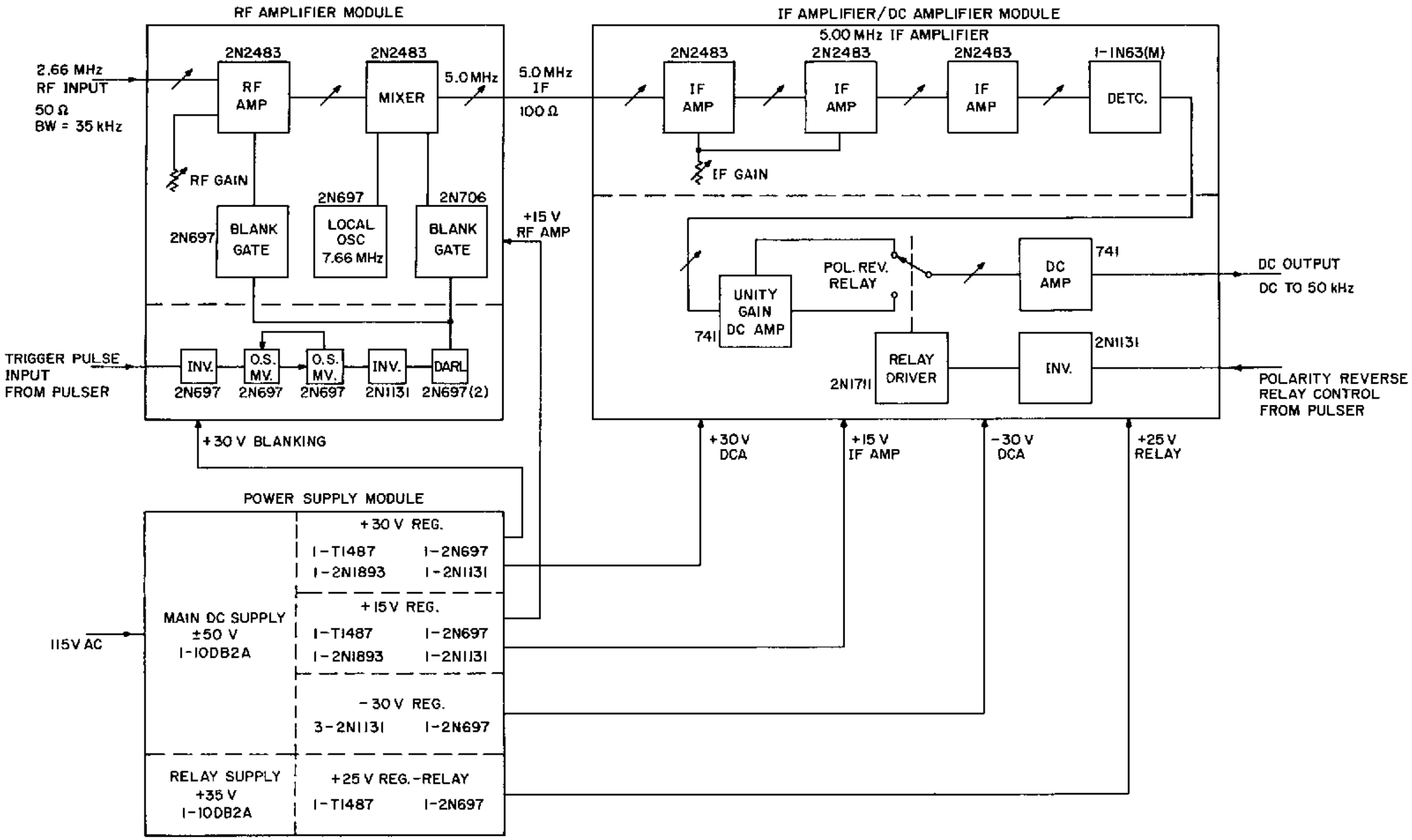

Figure 3.5 Block diagram of the revised receiver used to operate with a PDP-15 computer.

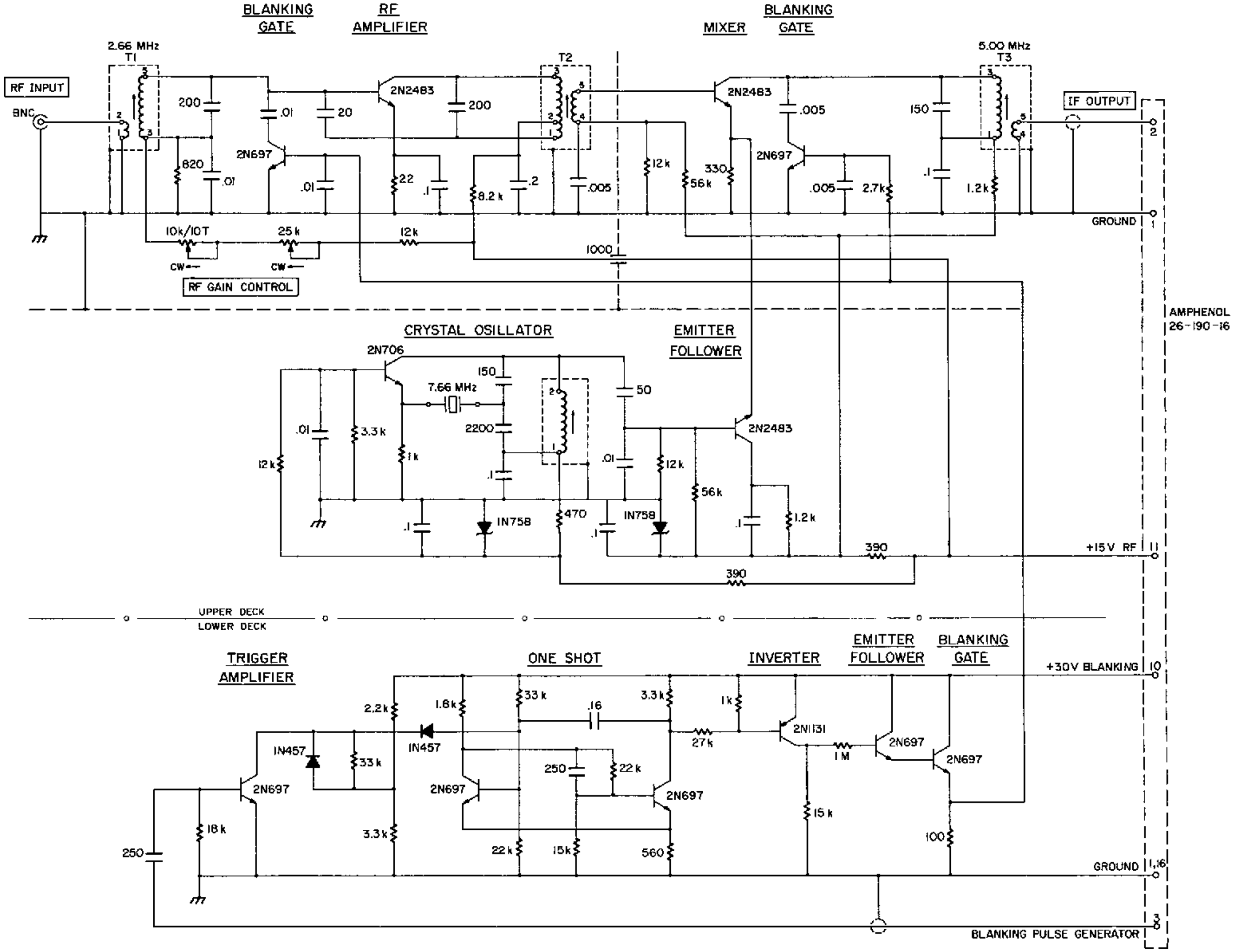

Figure 3.6 The RF amplifier module for the receiver.

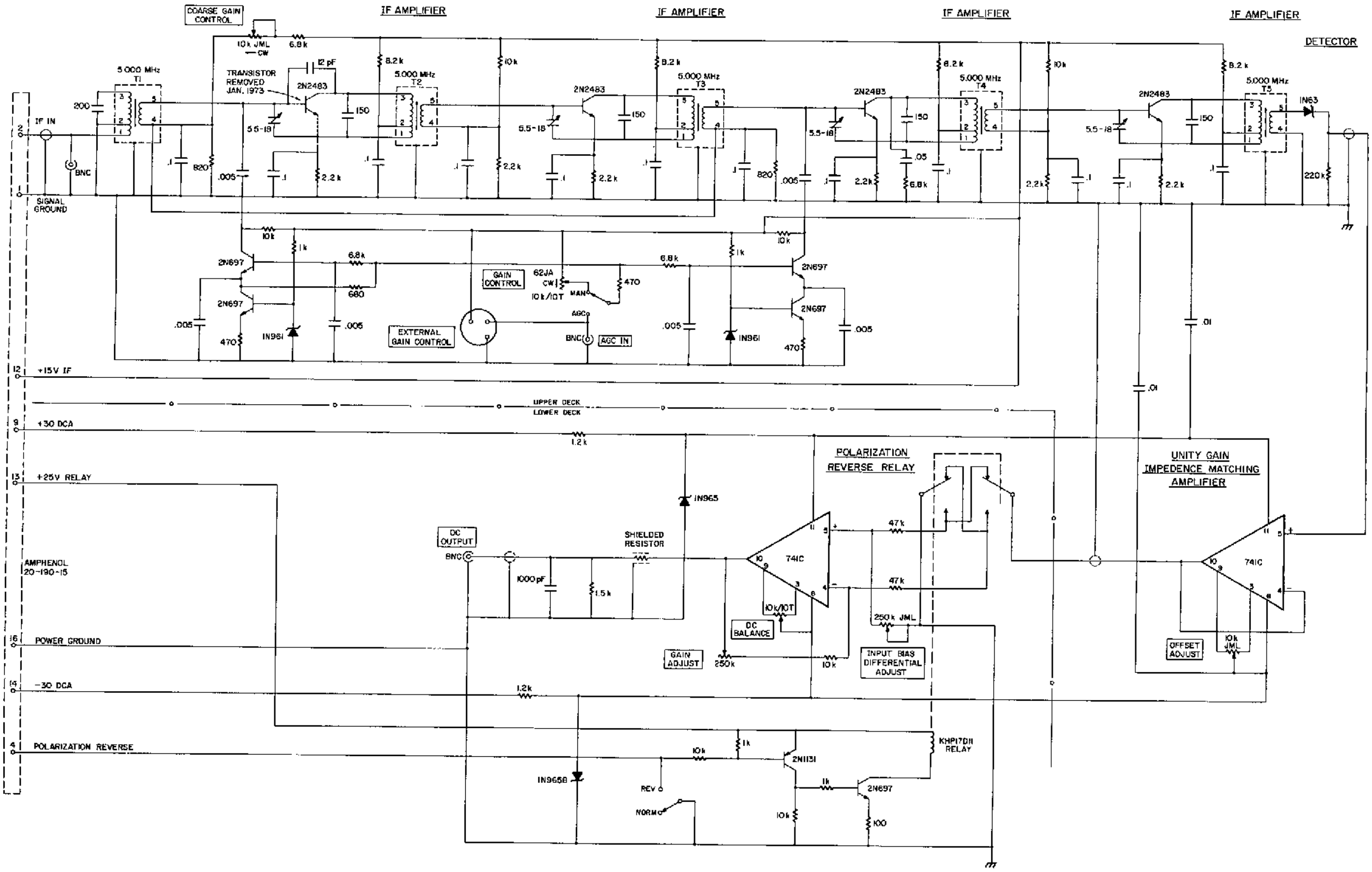

Figure 3.7 The IF and DC amplifier module.

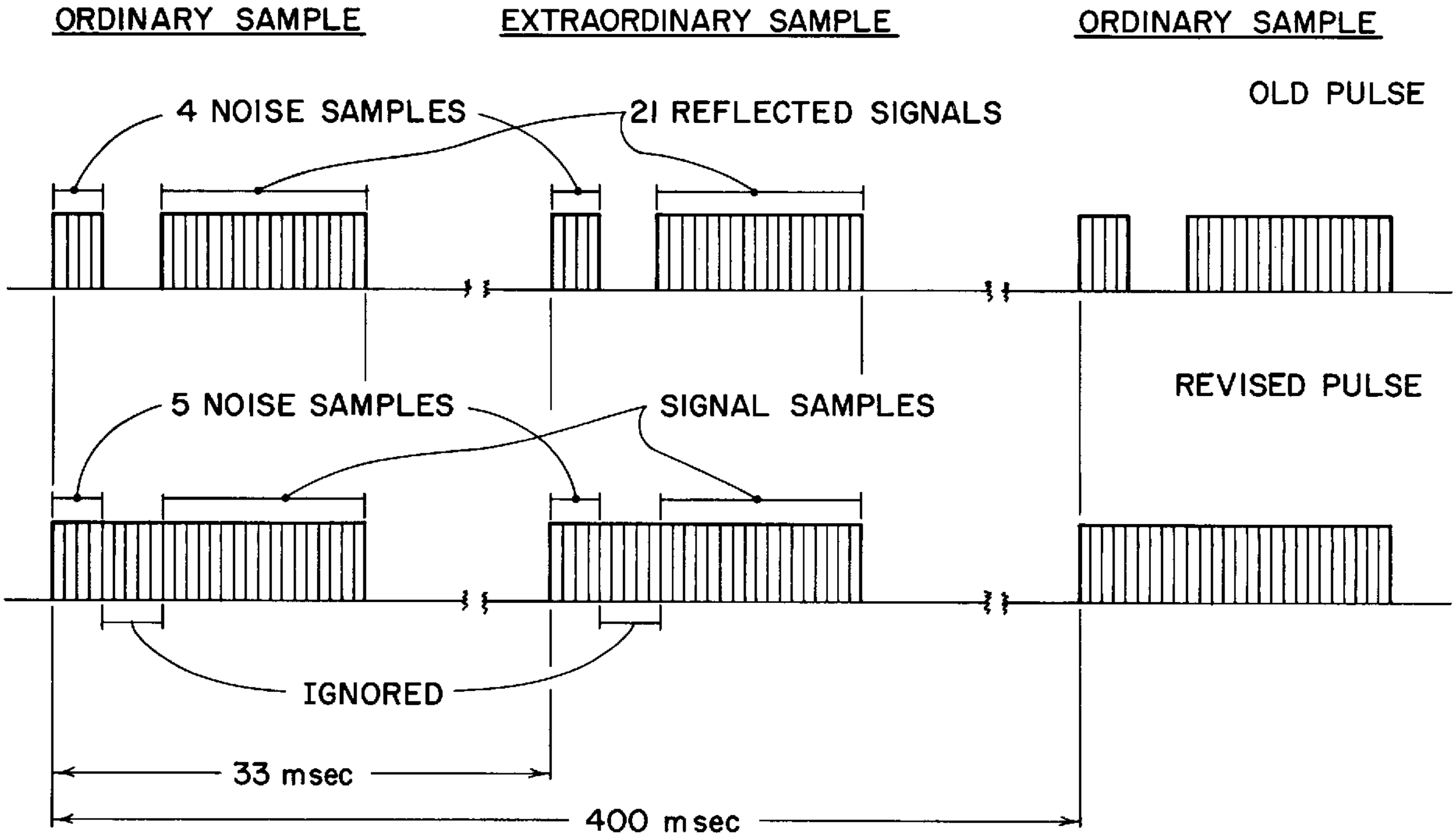

Figure 3.8 The encode pulses as set up by *Birley and Sechrist* [1971] used to collect data during the eclipse, and the revised encode pulses used by the present programs.

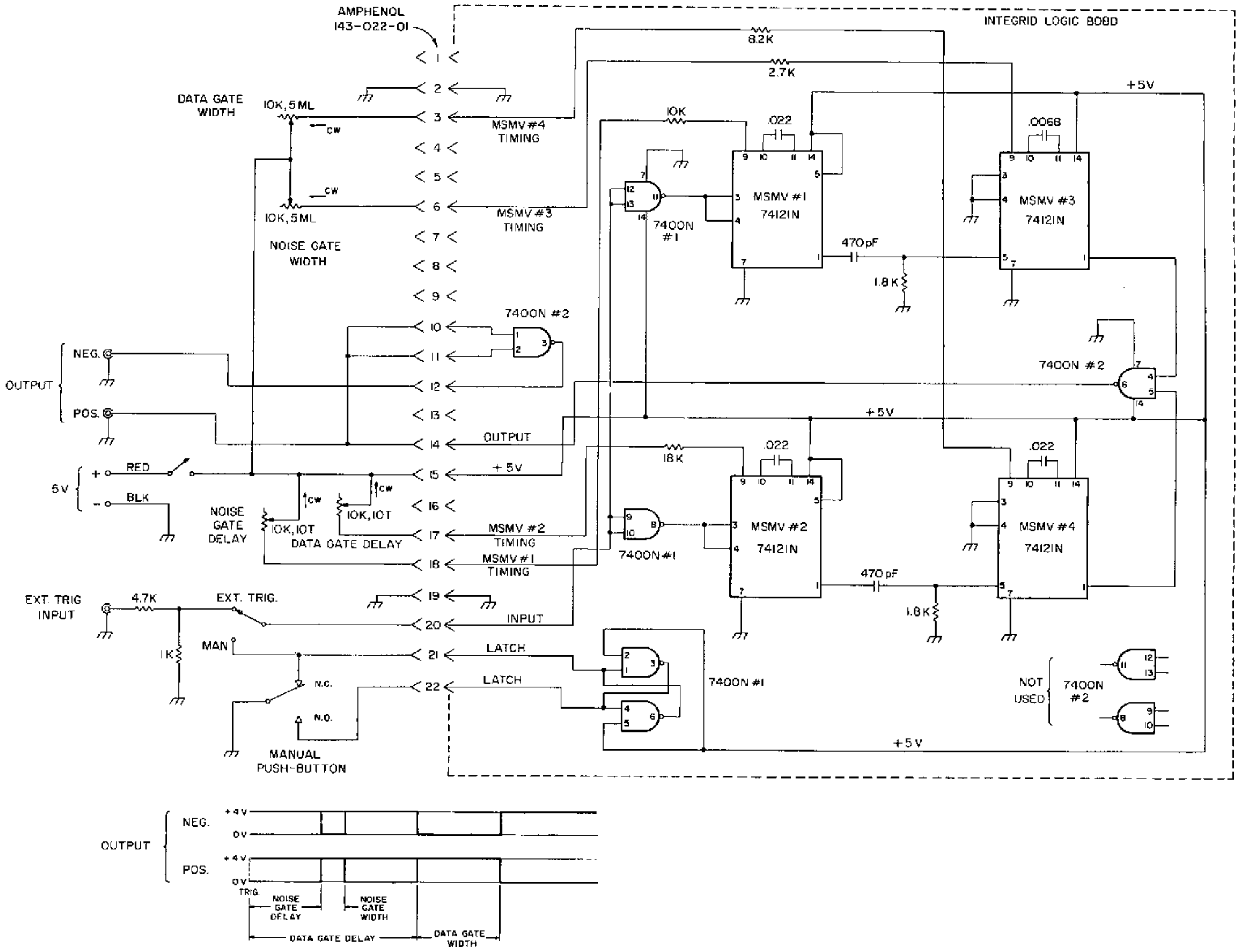

Figure 3.9 The encode pulse circuitry used to produce the former and present encode pulses.

to the 150 μsec, the set was from an extradordinary radio wave and is rejected. Othewise, both sets are accepted and the computer is synchronized with the pulser. This process is done only when the computer has a possibility of being out of synchronization with the pulser which are:

1) Beginning of every file
2) After the transfer of a block of data to disk
3) After collection is stopped and restarted by console control switch
4) During a timing error (no longer a terminal error, see Section 3.3)
5) When the computer "forgets to read" (discussed in Section 3.3)

The second change is to account for the nonlinearity of the receiver as seen in Figure 3.10 and was initially set up to adjust the data during processing [*Wiersma and Sechrist*, 1972]. Due to the time needed for the calibrating operation (approximately a half day), the computer is used which increases the speed of the process while making it possible to account for inaccuracies in the analog to digital converter. This process takes about 40 minutes (including 30 minutes for the receiver warm up). The adjustment to the data is done by using a table look-up method in the collection programs. Since the data stored on the disk are linearized data, the table is not needed after the collection is done and can be deleted after all the data are stored. The method is to convert the A/D converter output to the corresponding normalized receiver input. This is done by injecting a CW signal of a known value using an attenuator with one dB increments and storing the output in the computer using the set up shown in Figure 3.11. Straight line segment approximations to the curve in Figure 3.10 are obtained as shown in Table 3.1. Using outputs from 0 to 511 the corresponding inputs are determined normalized to 511 maximum, stored in a table as shown in

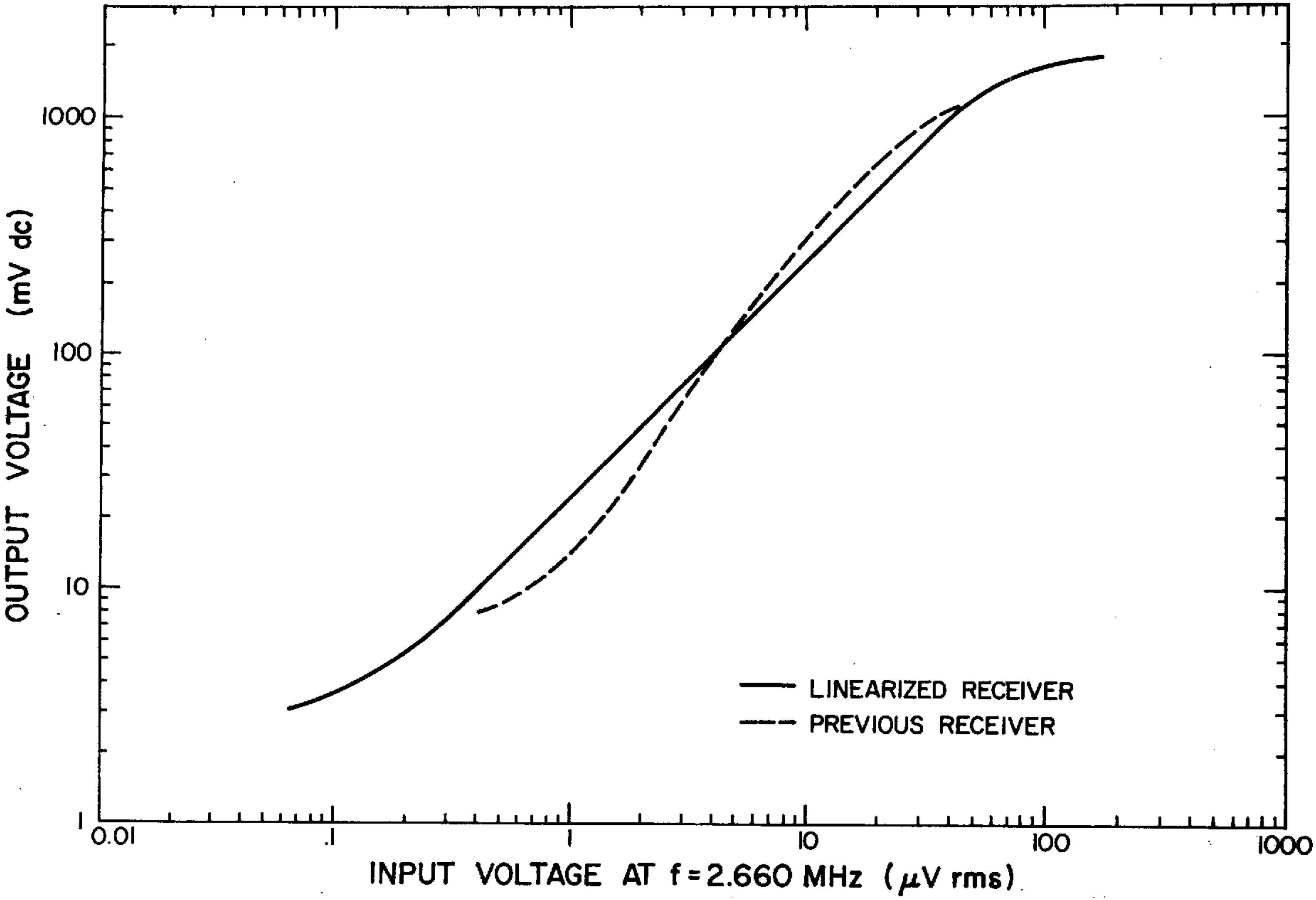

Figure 3.10 Graphs of the input versus output of the receiver used for eclipse data collection (old receiver) and the receiver presently being used. The input and output values have been normalized to the maximum of the A/D converter (511).

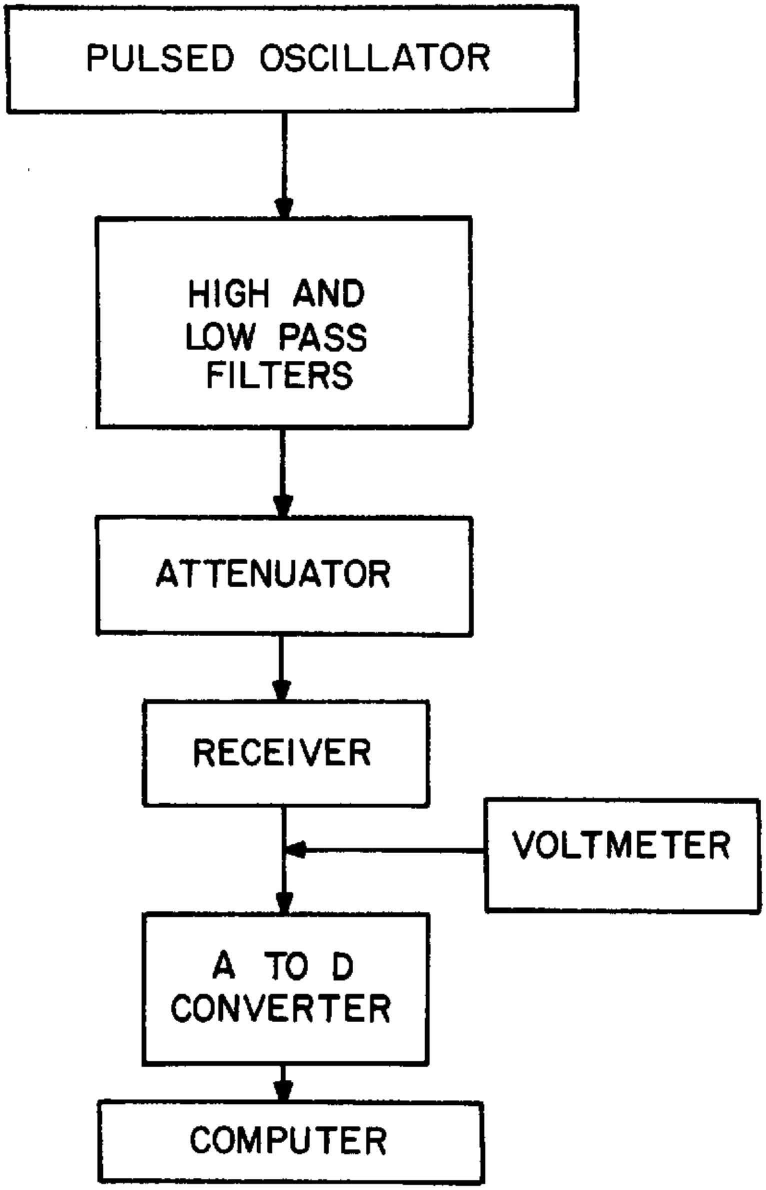

Figure 3.11 The wiring diagram used to calibrate the receiver. The voltmeter is used in setting the initial signal level prior to calibrating.

Table 3.1

Straight line segment approximation to the relationship of receiver input to receiver output.

Slope	Input	Output	Attenuation Used
S(1)=50.906	TU(1)= 0.000	TUO(1)= 4.786	99DB
S(2)= 3.912	TU(2)= 5.620	TUO(2)= 4.896	45DB
S(3)= 2.394	TU(3)= 6.310	TUO(3)= 5.073	44DB
S(4)= 7.762	TU(4)= 7.080	TUO(4)= 5.394	43DB
S(5)= 9.151	TU(5)= 7.940	TUO(5)= 5.505	42DB
S(6)= 2.576	TU(6)= 8.910	TUO(6)= 5.611	41DB
S(7)= 4.190	TU(7)= 10.000	TUO(7)= 6.034	40DB
S(8)= 1.983	TU(8)= 11.220	TUO(8)= 6.326	39DB
S(9)= 2.255	TU(9)= 12.590	TUO(9)= 7.016	38DB
S(10)= 1.676	TU(10)= 14.130	TUO(10)= 7.699	37DB
S(11)= 2.576	TU(11)= 15.850	TUO(11)= 8.726	36DB
S(12)= 1.946	TU(12)= 17.780	TUO(12)= 9.475	35DB
S(13)= 1.356	TU(13)= 19.950	TUO(13)= 10.590	34DB
S(14)= 1.754	TU(14)= 22.390	TUO(14)= 12.389	33DB
S(15)= 1.381	TU(15)= 25.120	TUO(15)= 13.945	32DB
S(16)= 1.012	TU(16)= 28.180	TUO(16)= 16.161	31DB
S(17)= 1.161	TU(17)= 31.620	TUO(17)= 19.560	30DB
S(18)= 0.969	TU(18)= 35.480	TUO(18)= 22.886	29DB
S(19)= 0.910	TU(19)= 39.810	TUO(19)= 27.353	28DB
S(20)= 1.113	TU(20)= 44.670	TUO(20)= 32.694	27DB
S(21)= 0.987	TU(21)= 50.120	TUO(21)= 37.591	26DB
S(22)= 0.771	TU(22)= 56.240	TUO(22)= 43.794	25DB
S(23)= 1.172	TU(23)= 63.100	TUO(23)= 52.693	24DB
S(24)= 0.708	TU(24)= 70.800	TUO(24)= 59.260	23DB
S(25)= 0.880	TU(25)= 79.430	TUO(25)= 71.451	22DB
S(26)= 0.832	TU(26)= 89.130	TUO(26)= 82.478	21DB
S(27)= 0.980	TU(27)= 100.000	TUO(27)= 95.540	20DB
S(28)= 0.789	TU(28)= 112.200	TUO(28)= 107.992	19DB
S(29)= 0.900	TU(29)= 125.900	TUO(29)= 125.364	18DB
S(30)= 0.816	TU(30)= 141.250	TUO(30)= 142.426	17DB
S(31)= 0.918	TU(31)= 158.490	TUO(31)= 163.559	16DB
S(32)= 0.766	TU(32)= 177.830	TUO(32)= 184.627	15DB
S(33)= 0.944	TU(33)= 199.530	TUO(33)= 212.947	14DB
S(34)= 0.913	TU(34)= 223.870	TUO(34)= 238.730	13DB
S(35)= 0.933	TU(35)= 251.190	TUO(35)= 268.645	12DB
S(36)= 1.030	TU(36)= 281.840	TUO(36)= 301.499	11DB
S(37)= 1.111	TU(37)= 316.230	TUO(37)= 334.895	10DB
S(38)= 1.260	TU(38)= 354.820	TUO(38)= 369.618	9DB
S(39)= 1.306	TU(39)= 398.110	TUO(39)= 403.979	8DB
S(40)= 1.471	TU(40)= 446.680	TUO(40)= 441.155	7DB
S(41)= 1.809	TU(41)= 501.190	TUO(41)= 478.202	6DB
	TU(42)= 562.340	TUO(42)= 512.000	5DB

Table 3.2, and placed on a storage device (normally a disk). The program DLOGF (given in the Appendix in MACRO language) reads Table 3.2 into the computer, and the table is used during collection of the received partial-reflection signal. Using the table, the MACRO subroutine LIN does the linearization of the numbers read from the analog to digital converter. The programs responsible for the formation of these two tables are TBFORL (FORTRAN IV), LINAP (FORTRAN IV), RADC (MACRO), and TTM (MACRO).

The system as it has been described was used to collect and process the partial-reflection data for the three-day eclipse period of July 9, 10, and 11, 1972. The rest of this chapter will describe further changes and developments of the system. These changes have been due to an increase of 16 K core memory, the addition of 2 disk units capable of storing 262.144 words each, and the changing from a single user monitor system to a background/foreground monitor disk system.

3.3 *Real-Time Data Storage and Automatic Processing*

A computer operates on its own timing system and if this timing system operates along with events outside the computer that affect the operation of the computer, then the computer is said to be operating in real time. For instance, if the computer reads in a set of 26 samples and is able to manipulate or process them before the next set of samples is read in, the computer is doing real-time processing; as opposed to saving the data on tape and processing it later, as done by *Reynolds and Sechrist* [1970]. With high-speed access on the disk (16 msec access time), the background/foreground system made possible real-time collection and processing of partial-reflection data. Due to the complicated timing, slow print-out, and the noise algorithm (discussed in Section 3.4), processing of the data is postponed until after the file is stored on the disk.

The background/foreground monitor system is a double monitor, multi-priority level, software system. The two monitors are separate software systems

Table 3.2

The output of the A/D converter are numbers between 1 and 511. The input for each output is given in this table.

1	1	1	1	1	7	11	12	14	16
18	19	21	22	24	25	26	27	28	29
30	31	32	33	34	35	35	36	37	38
39	40	41	41	42	44	45	46	46	47
48	49	50	51	52	52	53	54	55	55
56	57	57	58	59	60	61	63	64	65
65	66	67	67	68	68	69	70	70	71
72	72	73	74	75	76	76	77	78	79
80	80	81	82	83	83	84	85	86	86
87	88	89	89	90	91	92	93	94	95
95	96	97	98	99	100	101	102	103	103
104	105	105	106	107	108	108	109	110	110
111	112	113	113	114	115	116	116	117	118
119	120	120	121	122	123	124	125	125	126
127	128	129	129	130	131	132	132	133	134
135	135	136	137	138	138	139	140	141	141
142	143	143	144	145	146	147	148	148	149
150	151	152	153	153	154	155	156	157	158
158	159	160	161	162	162	163	164	165	165
166	167	167	168	169	169	170	171	172	172
173	174	174	175	176	176	177	178	179	179
180	181	181	182	183	184	185	186	186	187
188	189	190	191	192	192	193	194	195	196
197	198	198	199	200	201	202	203	204	204
205	206	207	208	209	209	210	211	212	213
214	214	215	216	217	218	219	219	220	221
222	223	224	224	225	226	227	228	229	229
230	231	232	233	234	234	235	236	237	238
239	240	240	241	242	243	244	245	245	246
247	248	249	250	251	251	252	253	254	255
256	257	257	258	259	260	261	262	263	264
265	266	267	268	269	270	271	271	272	273
274	275	276	277	278	279	280	281	282	283
284	285	286	286	287	288	289	290	291	292
293	294	295	296	297	299	300	301	302	303
304	305	306	307	308	309	310	311	312	313
314	315	316	317	318	319	320	321	322	323
324	325	326	327	328	330	331	332	333	334
335	336	338	339	340	341	342	343	344	346
347	348	349	350	351	353	354	355	356	357
358	359	361	362	363	364	365	366	368	369
370	371	372	374	375	376	377	378	379	381
382	383	384	385	387	388	389	390	391	393
394	395	396	397	399	400	401	402	403	404
406	407	408	410	411	412	414	415	416	418
419	420	422	423	424	426	427	428	430	431
432	434	435	436	438	439	440	442	443	444
446	447	448	450	451	452	454	455	456	458
460	461	463	465	466	468	470	471	473	474
476	478	479	481	483	484	486	488	489	491
493	494	496	498	499	501	502	504	506	507
509	511								

sharing the same hardware with programs operating in the foreground system having priority. Each system has 8 automatic priority (API) levels and a mainstream level. There are four hardware levels which have highest priority. The software levels are labeled 4, 5, 6, 7, and 0 where 4 is the highest and 0 is the mainstream, the lowest. When a program initially starts running in either background or foreground, it begins on mainstream. Certain commands require a special subroutine called a real-time subroutine and is designated a priority level from 0 to 4 and stops all operation on lower priority levels (background is lower than foreground) until it exists from the level or performs an I/O operation.

With this system the partial-reflection collection and processing programs as mentioned could operate in real time, but due to several problems in the processing of data, the data could not easily be saved except in processed form. The solution used is to collect one file of data and process that file while the next file of data is being collected. After each file is collected, the operator is told what the next attenuator setting is. The collection program also checks the setting of the switches on the console to allow the operator to control parts of the collection program. Switch 0 acts as an on/off switch which causes collection to stop collecting and wait in a loop if set to 1. Switch 1 allows the background system to share the collection and processing storage device (1 disk) if the switch is set to 1. This sharing is necessary if the collected files are to be stored on DECtape. Switch 3 allows the processed data which are printed out onto the teletype to also be punched onto paper tape if the switch is set to 0. This option is presently used to allow for later plotting of the data using a programmable calculator. Switches 2 and 5 are not used at the present. The rest of the switches are used for determining the length of each file (default length is 513 pairs of sets of

26 numbers). The time of day is determined by using the clock within the computer to give the time in hours and minutes.

The flow diagram of the programs is shown in Figure 3.12. The programs are loaded into the computer and the computer's clock is set to the time of day. The operator is given the option of calibration of the receiver. The linearization table is stored on a disk and some initial information is read in. If the table read in is erroneous the operator must re-do the calibration procedures. The collection is started on priority level 6 and processing waits for the first file to be collected. After collection of the number of sets of samples set on the console switches and the operator changes the attenuator setting, the second file is collected while the first is processed and printed out. This process continues until stopped by the operator. Information used to calculate the noise threshold as described in Section 3.4 is transferred to the processing program after each file is collected and is not stored on the disk. The processing program therefore must remain faster than the collection or this information will be lost.

The processing of files involves rejecting sets of samples that are too noisy (discussed in Section 3.4), summing the squares of unsaturated data, subtracting off the sum of the squared acceptable data, and taking the square root. The resulting data are two sets of 21 samples, one of ordinary modes (A_o) and one of extraordinary mode (A_x) radio waves. This process is done in the main processing program PROC (given in the Appendix). The electron densities are calculated in CALC2 which is discussed in Section 3.5. The results are typed out on the teletype in tabular form as shown in Table 3.3.

The first line of the print-out of processed data is the heading. This gives the time the collection of the file stopped, the date, the reason for the run, and the attenuator setting for the file. The next line contains the noise threshold and the square of the multiplying constant used in the

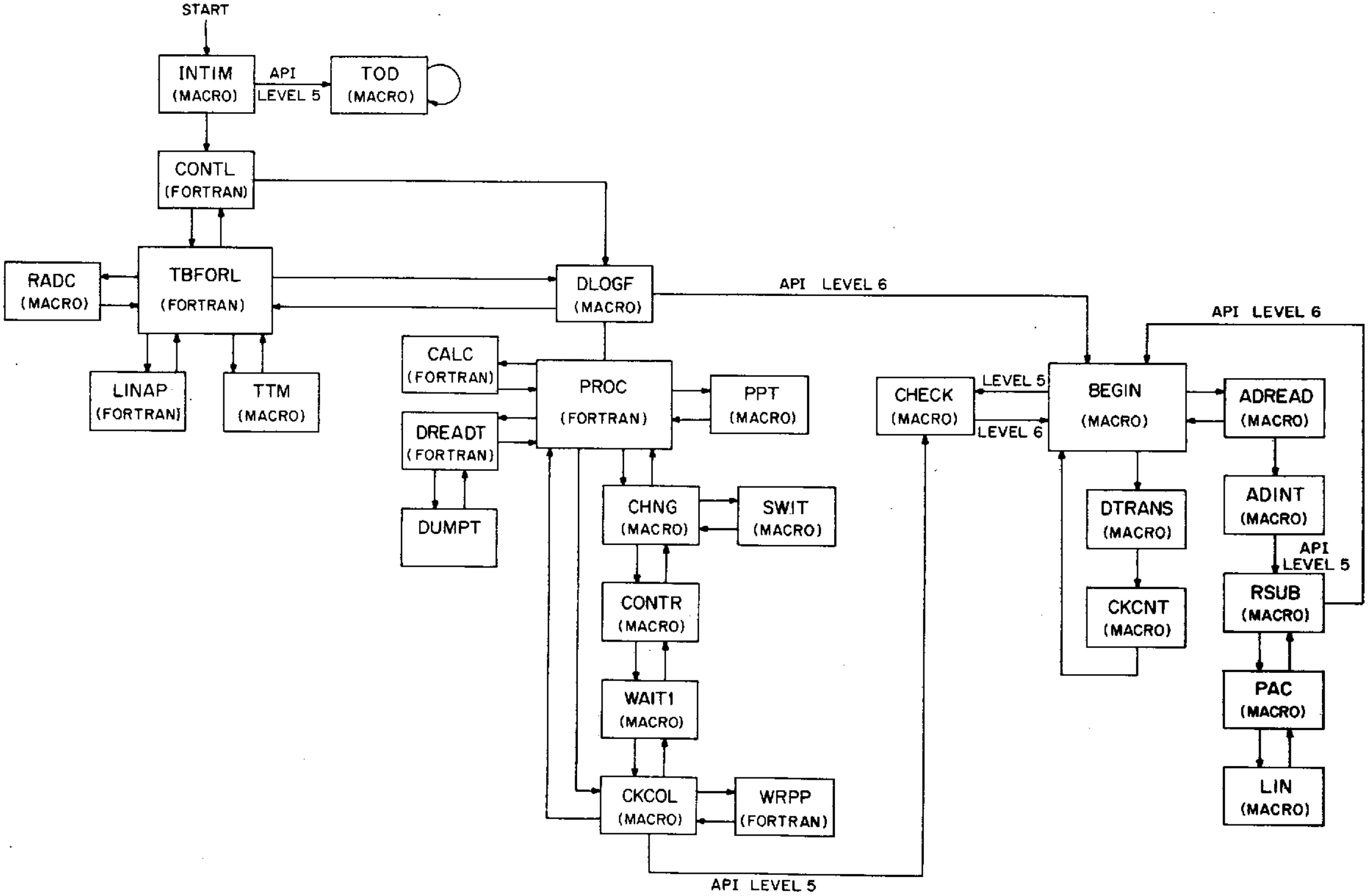

Figure 3.12 A diagram of the control flow of the partial-reflection programs. The programs operate on the API level of the preceding program unless otherwise stated. The collection and processing program operated in parallel with the collection programs operating on API levels 5 and 6, whereas everything else operates serially.

TABLE 3.3

Results of CALC2

```
1215     8-14-73        DAILY  RUN                       10DB


MAX. ALLOW. NOISE=   19.8   MULT. CONST.=  9.610

O-NOISE AV.(1)      19.9    (2)       11.6
X-NOISE AV.(1)      12.0    (2)        7.5


 513 SAMPLES          44 REJ.(NOISE)

  REJ.
(N.+SAT.)     HEIGHT     AV. AO   AV. AX   AX/AO     ED

     44        60.0        6.9      4.1     0.59
                                                   5173.
     44        61.5        9.9      2.0     0.20
                                                      0.
     44        63.0        3.3     -2.1     0.00
                                                      0.
     44        64.5        3.8      4.6     1.22
                                                   -247.
     44        66.0        7.8     11.5     1.46
                                                     59.
     44        67.5       14.4     22.0     1.53
                                                    105.
     44        69.0       22.5     34.9     1.55
                                                    199.
     44        70.5       30.6     44.9     1.47
                                                    271.
     44        72.0       30.8     40.0     1.30
                                                    339.
     44        73.5       36.2     39.0     1.08
                                                    298.
     44        75.0       54.2     49.8     0.92
                                                    340.
     44        76.5       59.1     45.1     0.76
                                                    502.
     44        78.0       58.9     34.1     0.58
                                                    416.
     44        79.5       57.5     27.3     0.48
                                                    682.
     44        81.0       56.5     20.2     0.36
                                                    778.
     44        82.5       52.7     14.4     0.27
                                                    623.
     44        84.0       46.6     10.8     0.23
                                                   3003.
     44        85.5       99.0     12.0     0.12
                                                    445.
     44        87.0      228.9     26.0     0.11
                                                   -694.
     53        88.5      307.2     38.0     0.12
                                                   -469.
     99        90.0      324.0     41.9     0.13
                                                      0.
```

maximum noise criterion discussed in Section 3.4. The next two lines are the ordinary and extraordinary mode noise before (number 1) and after (number 2) rejections due to excessive noise. The next line gives the number of pairs of sets of 26 samples collected and the number of these pairs rejected due to saturation. The first column of the table is the number of rejections due to both saturation and excessive noise for each height. The next column gives the height of the reflected signals for each row. The next two columns give RMS of the ordinary (A_o) and extraordinary (A_x) signals. The fifth column gives the ratios of extraordinary partial reflections to ordinary partial reflections from the fourth and third column respectively. The last column gives the electron density for between the heights. The last electron density is given as zero since only one height is available to calculate it.

The present method of collection and processing of partial-reflection data is fast, efficient, and easy to operate, but two problems needed to be removed. The increase of input/output operations have increased timing errors which are discussed by *Birley and Sechrist* [1971], and the A/D converter sometimes fails to respond to read commands.

The A/D converter transfers data to the computer using multicycle block transfer as described by *Birley and Sechrist* [1971]. The process is a three cycle operation for each word transferred. After each transfer, the A/D converter interface is tested for synchronization. If the timing between the interface and the I/O processor is altered, transfer is stopped resulting in a timing error. With the present system, this error can result from hardware malfunction or excessive I/O operation occurring. If the latter is the reason, the problem is only temporary and can be remedied by issuing another read. Care is taken to keep the computer synchronized with the pulser. If the error is a hardware problem, the condition will not clear up and collection must stop. The error will usually occur when data are being collected, processed data are being printed out, and a tape is being copied onto the disk in background, all simultaneously.

The second problem has to do with the A/D converter's interface refusal to transmit data. The problem has been traced to failure in the A/D converter interface logic. The collection program will issue an A/D converter read, but not receive control back and no data are transferred. This problem occurs only with the background/foreground system and it occurs infrequently (once in about every 10,000 read commands). One solution is to issue a double read, but the problem could still occur. The solution used is for the processing to check for this stoppage, restart the collection in an orderly fashion if it has stopped and to ring the teletype bell to let the operator know of the stoppage. This solution does not prevent the failure of the A/D converter interface to transfer data, and the problem will have to be removed for faster ratio of collection, but presently the operator need not be concerned with this problem. The rest of the data is unaltered by this problem.

3.4 *Noise Rejection*

The partially reflected radio waves from the *D* region are usually small in amplitude on the order of 10 to 1000 mvolts at the output of the 80 dB gain receiver. Noise amplitudes vary between 30 to 1000 mvolts. For the purpose of the noise algorithm, noise is considered to be any interference which is part of the receiver output signal that is not attributed to the partially reflected waves from the vertically transmitted pulse. This noise is divided into two types: background noise and noise bursts. Background noise is noise caused by the reciever (14±3 mV) and general atmospheric noise which is always present (40±10 mV). Noise bursts are caused by lightning and other radio transmitters, and the amplitude of this noise is dependent on the location of the source. Lightning noise will usually last for the duration of one encode pulse while noise due to other transmitters will last for at least 1/2 second which is several encode pulses (see Figure 3.8) and the noise will be increased usually by 10 to 1000 mvolts. Both types of noise are rejected in the processing program PROC(FORTRAN IV) as shown in the block diagram of this program in Figure 3.13.

Data are collected in pairs of sets of 26 numbers. Each set contains 5 noise samples and 21 samples of partially reflected signals. Each pair contains a set of ordinary mode samples and a set of extraordinary mode samples. In PROC a noise threshold is determined and the square of this multiplied by five is compared to the sum of the squares of the five noise samples of each set. This method of comparison is faster than comparing the RMS of the noise as set up by *Birley and Sechrist* [1971] since square root operations take approximately 1 msec and squaring takes 70 μsec on the PDP-15, and the squaring need only be done once per file. If the noise of either mode is greater than the noise threshold, both sets of 21 signal samples are rejected and the next pair of sets are tested. If the noise of both modes is less than this threshold, the noise of both sets are considered acceptable and saved for later processing. The partially reflected signals with acceptable noise for each mode are checked for A/D converter saturation (.997 volts receiver output) at each height. If either of the two samples (one of each mode) is saturated at a height the two samples are rejected; otherwise the data are considered acceptable. This processing of pairs of 26 samples continues until the end of the file is reached. After the file of collected data has gone through this processing, the average of the sum of the squared acceptable noise for each mode is subtracted from the average of the sum of the squared acceptable partially reflected samples of the same mode at each height, and the square roots are printed out as shown in Table 3.3 and as described in Section 3.3.

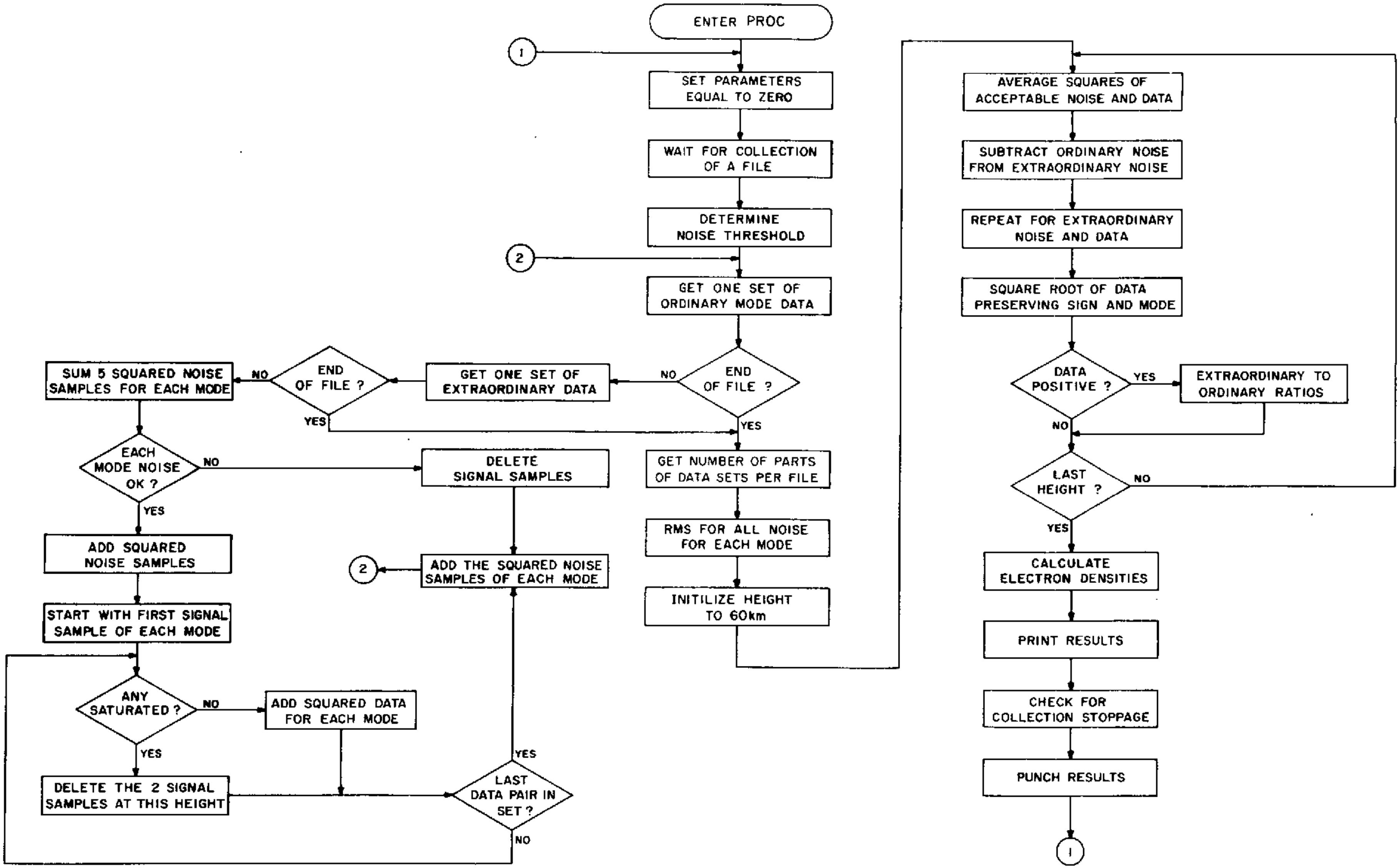

Figure 3.13 A flow chart of the processing program PROC.

Originally, the noise threshold was determined by the operator typing in a value chosen by him as seen in the program PROC73 in the Appendix. This was later changed to an automatic determination based on the attenuator setting used as given at the beginning of a run. This method did not account for the day-to-day variation in noise nor in an erroneous attenuator setting. The noise threshold value is presently determined by the following equation:

$$M = (K(\sum_{0}^{45} N)/45)^2 \tag{3.1}$$

where M = maximum allowable noise value

K = arbitrary constant

N = certain noise samples collected as explained in the following paragraph.

In the collection programs RSUB and LIN, the maximum and sum of each group of 45 noise samples are stored, and the maximum values are compared. The sum of the group with the lowest maximum value is transferred to the processing program PROC and is used in equation (3.1). The constant K has been chosen by trial and error, and values between 2.5 and 3.5 seem to give the best results (equation 3.1 is being used).

Other algorithms have been tried, but none seem to give any obvious improvement in the resulting electron densities. One method is to split 5 noise samples collected with each set of data into 2 for comparison with the noise threshold value and 3 subtracted from the reflected signals. This method works on the theory that the noise within the 5 noise samples is not the same amplitude as the noise within the 21 data samples for each set of 26 data samples, but is statistically the same over the number of samples collected for one file. With the present system, when the number of rejections due to noise is large, (greater than 200 out of 513 pairs of sets of samples), the noise within the noise gate is restricted to a lower level than the noise in the data frame. Therefore, the noise in the data frame would not be completely subtracted off; as it would be with splitting the noise samples. The application of this technique using 4 noise samples showed no improvement in the results. Two possible causes are too few noise samples being used and the noise samples being too close together.

Another method has been developed and tested by *D. R. Ward* [private communication]. A CW signal is inputed into the receiver along with the received data from the antenna. The noise and partially reflected signals are each defined as A cosθ; where A is the amplitude and θ is the phase. The noise is assumed to be random while the partially reflected signals are assumed to have only a small variation between two sets of samples. Using an algorithm developed by *D. R. Ward* [private communication], the phase and the amplitude of the noise portion of each signal average to zero while the phase and amplitude of the signals do not. This method is used to reject the noise from the partially reflected signals at each height. This method fails to reject interference caused by other transmitted signals since this type of noise does not have random phase. *D. R. Ward* [private communication] has obtained useful electron-density profiles from the method but generally found no improvement over the present system. Further study and development of either method may improve the processing and should not be discarded.

3.5 *Converting* A_x/A_o *Ratios to Electron-Density Profiles*

The partial-reflection programs assume a constant collision frequency for each height with seasonal variation. The values used were determined from the following equation [*Birley and Sechrist,* 1971]:

$$\nu_m = Kp \qquad (3.2)$$

where K = constant = 7.3 x 10^5

p = pressure in pascals

ν_m = collision frequency in $\sec^{-1}$

The pressures used are from the mean atmospheric model from COSPAR International Reference Atmosphere (1965) with seasonal variations given by U. S. Standard Atmospheric Supplements (1966). Using these pressures, experimentally the values calculated for K vary by as much as 2 x 10^5 [*Lodato and Mechtly,* 1971]. The seasonal variations in the collision frequency (Figure 3.14) can vary by as much as 20%. This 20% variation in ν_m can cause the calculated $[e]$ to vary by a factor of 1.2. The electron densities are calculated using the refractive index equation given by *Sen and Wyller* [1960] and several approximations as discussed by *Pirnat and Bowhill* [1968]. The resulting equation given by *Reynolds and Sechrist* [1970] is:

$$[e] = \ln\{ \left[(A_x/A_o) \;/\; (R_x/R_o) \right]_{h_1} \;/\; \left[(A_x/A_o) \;/\; (R_x/R_o) \right]_{h_2} \}/\,\mathrm{FD} \qquad (3.3)$$

$$\mathrm{FD} = (5\Delta h e^2/2cm\varepsilon_o\nu_m) \quad \{\zeta_{5/2}\left[(\omega-\omega_L)/\nu_m\right] - \zeta_{5/2}\left[(\omega+\omega_L)/\nu_m\right]\} \qquad (3.4)$$

where

$$\zeta_y(x) = \frac{1}{y!}\int_o^\infty \frac{\varepsilon y}{\varepsilon^2+x^2}\, e^{-\varepsilon}\, d\varepsilon$$

$\varepsilon = mV^2/2kT$

$[e]$ = electron density

e = electron charge = 1.6 x 10^{-19} C

m = electron mass = 9.1 x 10^{-31} kg

ε_o = permittivity of free space = 8.85 x 10^{-12} F m^{-1}

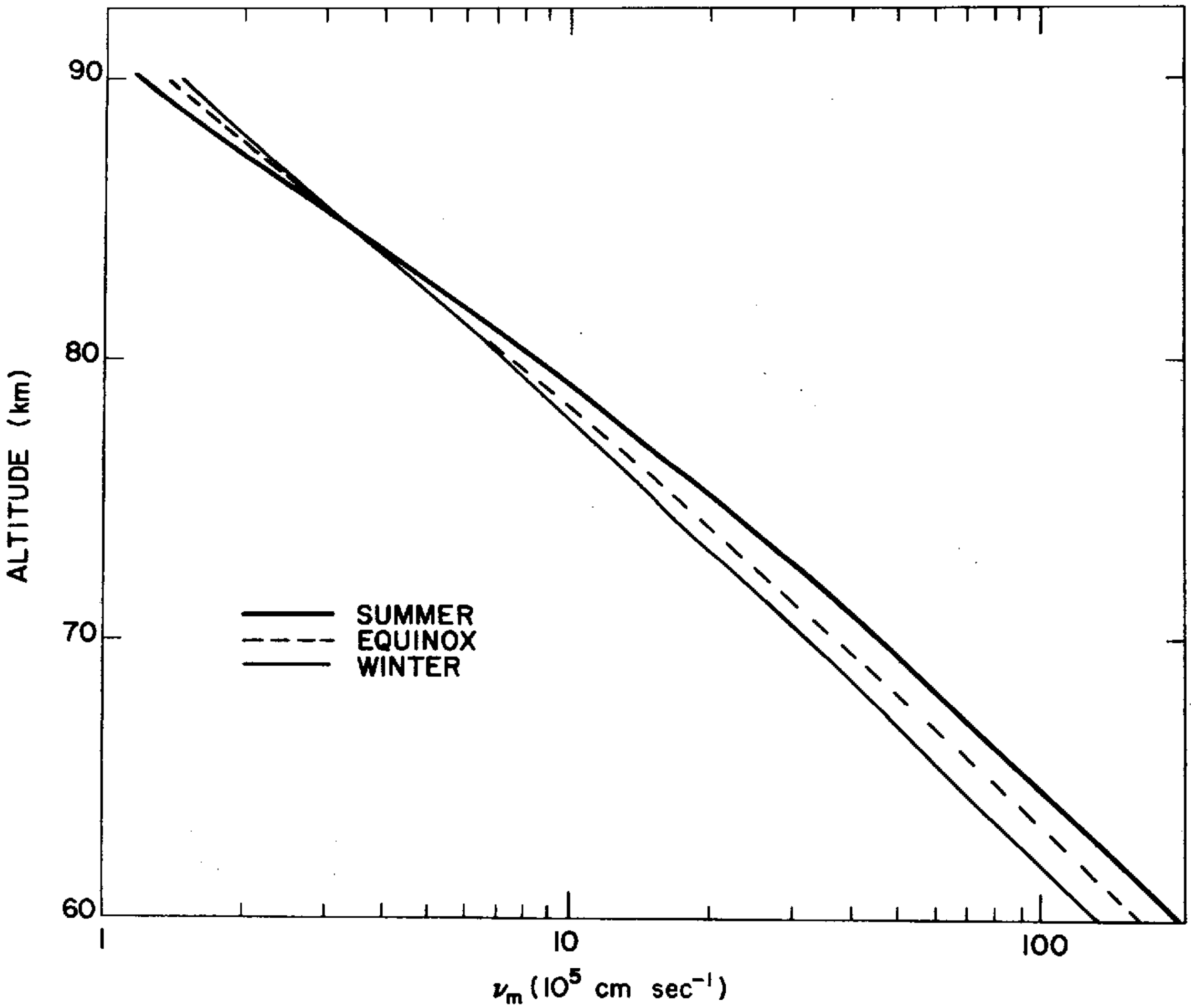

Figure 3.14 The collision frequencies used in the program CALC to obtain electron-density profiles.

ω = angular frequency of the transmitted wave

ω_L = gyro-frequency of the electron

h_1 = lower height

h_2 = higher height

$\Delta h = h_2 - h_1$

k = Boltzmann constant = 1.38×10^{-23} J° K^{-1}

T = temperature

V = electron velocity

R_o = ordinary mode reflection coefficient

R_x = extraordinary mode reflection coefficient

This equation required a set of collision frequency constants which are given in the program CALC (FORTRAN IV). The ratio $(R_x/R_o)_{h_2} \Big/ (R_x/R_o)_{h_1}$ and FD (equation (3.4)) are calculated in ELDEN (FORTRAN IV). CAL2 (called by PROC) uses these values (which vary only with ν_m) as constants for each pair of heights to calculate the electron densities according to Equation (3.5)

$$[e] = \ln\Big(\mathrm{RATIO2} \times (A_x/A_o)_{h_1} \Big/ (A_x/A_o)_{h_2}\Big)/\mathrm{FD} \qquad (3.5)$$

where RATIO2 = $(R_x/R_o)_{h_2} \Big/ (R_x/R_o)_{h_1}$.

This method is used to reduce the amount of core memory required and increase speed of execution of the program. A new CALC2 can be obtained by revising the collision frequencies and running the program CALC which writes the program CALC2. The electron densities are printed out as shown in Table 3.3 and described in Section 3.3

3.6 *Equipment Testing*

The equipment needs to be tested periodically to determine if it is in operating order. The transmitter is tested by observing and keeping a log of the voltage and current at various locations via meters and an oscilloscope. The antennas are tested by transmitting and receiving signals at various times during the day. At noon the extraordinary signal should be absorbed and at night the ordinary signal should be absorbed. By transmitting and receiving ordinary and extraordinary signals as described in Progress Report 73-1 [*Edwards*, 1973], the phase and attenuation of each antenna of each array can be set and checked for possible damage. This process is also a partial check for the transmitter and receiver. A spot check of 30 dB difference in ordinary and extraordinary reflections from the *E* region at noon is done on a daily basis.

The program CHECK (FORTRAN IV) has proved valuable in checking the receiver and the analog to digital converter. CHECK performs a modified dump of the A/D converter as read by the computer. If the number 31 is typed, the output is in the form of partial-reflection data (ordinary and extraordinary pairs), patterned after the new encode pulse shown in Figure 3.8. If any other number is typed in an average of that number rounded to the next higher multiple of 50 is printed out. The 31 pairs of samples are printed out in millivolts only, while the averages are printed out in millivolts and as represented in the A/D converter. This program has had many applications; it showed the blanking gate on a new receiver to be too long. It was used to calibrate the A/D converter using an input from a standard source. Table 3.4 shows the accuracy of the A/D converter as the standard voltage source was varied from 1.0 volts to .1 in .1, .01, and .001 volt increments. It was used in comparing the paper punch system set up by *Reynolds and Sechrist* [1970] with the computer storing method presented in Section 3.3. CHECK has also been used to determine the number of samples required to have less than 10% error due to noise (at least 100 samples are required). The program is easy to operate and has become important in testing and checking the receiver and the analog to digital converter.

Table 3.4

The output of the A/D converter using a calibrated input source

ADC Output		Input
Average	Voltage	Voltage
511.204	998.444 mV	1000 mV
460.558	899.527 mV	900 mV
409.625	800.050 mV	800 mV
358.528	700.250 mV	700 mV
307.057	599.720 mV	600 mV
256.020	500.040 mV	500 mV
205.252	400.883 mV	400 mV
154.082	300.941 mV	300 mV
102.787	200.756 mV	200 mV
51.076	99.758 mV	100 mV
46.349	90.525 mV	90 mV
40.843	79.772 mV	80 mV
35.208	68.766 mV	70 mV
30.844	60.242 mV	60 mV
25.769	50.330 mV	50 mV
19.976	39.016 mV	40 mV
15.022	29.339 mV	30 mV
10.200	19.922 mV	20 mV
4.830	9.433 mV	10 mV
3.857	7.533 mV	9 mV
3.233	6.315 mV	8 mV
3.010	5.880 mV	7 mV
2.847	5.560 mV	6 mV
2.443	4.771 mV	5 mV
2.054	4.012 mV	4 mV
1.404	2.743 mV	3 mV
0.659	1.286 mV	2 mV
0.125	0.244 mV	1 mV
0.010	0.020 mV	0 mV

3.7 *Future Development*

Several improvements are being made to the system. A new receiver is being made using a linear detector and new RF and IF stages to reduce the receiver noise. Figure 3.10 shows a comparison of the input versus output between the new receiver and the old one. With no input signal, the noise level of the new receiver is 2.5 mV and the level of the older receiver is 14 mV. The circuitry and discussion of it are given in the Aeronomy Progress Report 73-1 [*Edwards*, 1973].

A digital input/output device is presently being sought which would improve the calibration time and free the operator for other tasks as well as simplify the operation of the system. The purchase of such a device would also reduce the amount of paper presently required.

Another asset would be a line printer. One could reduce the processing time by at least half and allow for more sophisticated processing (with possibly better noise rejection) if such a line printer were purchased.

As mentioned by *Birley and Sechrist* [1971], an increase of transmitter power is also needed. This would improve the signal-to-noise ratio and give better data below 70 km.

The noise problem should be studied more carefully. Perhaps a combination of the method discussed in Section 3.4 would improve the results. Another possibility would be to reject extremely low values of reflected signals.

An additional program to transfer collected data to tape would be helpful. The original programs set up by *Birley and Sechrist* [1971] saved data on tape for future processing. With the present system, collected data can be stored on tape by using a system program called PIP. This requires knowledge in operation of the computer, and the transferring of files can get complicated.

4. EXPERIMENTAL RESULTS

This chapter describes the results from partial-reflection data which was collected and processed by the computer on July 9, 10, and 11, 1972. A solar eclipse occurred on July 10, 1972. The obscuration function shown in Figure 1.1 shows the first contact to be at 1319 CST and the last contact to be at 1536 CST with 60% of the solar disk obscured. The data were collected from 1200 to 1700 CST to show the effects of the solar eclipse on the electron density and collected between the same times on July 9 and 11 to be used as control data. Data were collected in blocks called files. Each file of data, consisting of 1026 sets of 26 numbers, was collected and stored on DECtape every 3.8 minutes. The signal prior to entering the receiver was attenuated with four attenuator settings (0, 10, 20, and 30 dB). Each file was collected beginning with the lowest attenuator setting of 0 dB with each subsequent file collected at the next attenuator settings; 10, 20, and 30 dB, respectively. This process was then repeated. This process was used to obtain the very small echoes as well as the very large ones. The files of data are divided into approximately 15 minute intervals, corresponding to the four attenuator settings.

The data between 1400 and 1430 on July 9 was lost due to an erasure of the disk before it could be processed. These data have been interpolated. The data from July 10 between 1200 and 1300 was erroneous and therefore has been eliminated from the results. The computer results were processed further combining the files with different attenuator settings.

4.1 *Reduction of Data*

Individual results shown in Figure 4.1 show valid electron densities but are limited height range; therefore, multiple attenuator settings were used to obtain usable data over a greater range of heights. The computer processes

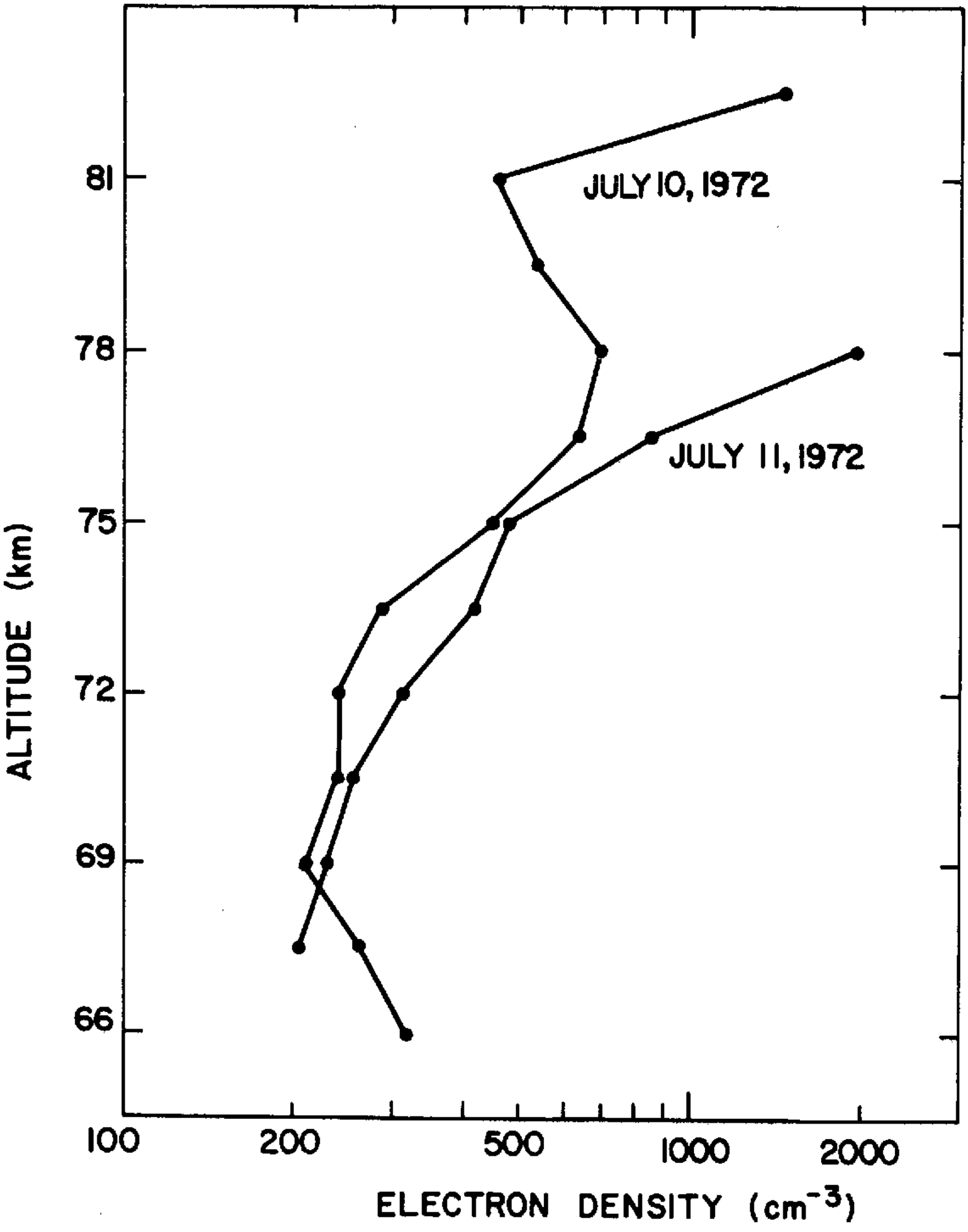

Figure 4.1 Comparison of electron-density profiles on July 10 and 11, 1972. The data were taken at 1432 CST with the attenuator set at 30 dB.

only one file at a time; therefore, further processing was necessary to combine four files corresponding to the four attenuator settings into one set of results. Three methods have been developed to accomplish this. The first method was originally used but problems developed in determining acceptable data and method two was used. Using method two, some acceptable data were being ignored and the 20 and 30 dB settings were found to give similar results. Therefore, method three was developed to utilize much of this acceptable data that were being ignored.

1. In method one, the results with the lowest attenuator setting (0 dB) were used for 60 km up to the height where 5% of the ordinary and extraordinary data was rejected due to saturation (see Section 3.4). The electron densities for the higher heights were obtained from the next higher attenuator setting under the same restrictions of saturations. This process continued until the last electron density was obtained. The results of this method seemed to be satisfactory except for above 81 km and below 66 km.
2. Method two is the same as method one, but accounts for inaccuracies in the receiver by rejecting electron densities that used A_x/A_o ratios that were less than .09. Electron densities were rejected also if the signal to noise ratio was less than 1. These two revisions eliminated much of the results below 65 km and above 85 km.
3. Method three is similar to method two except for the way the multiple attenuators are combined. The electron densities are considered acceptable if the A_x/A_o ratios for both heights are greater than .08, the signal to noise ratio is above 1 for both heights, and the rejections due to saturations were less than 5% for both heights used to calculate the

electron density. If more than one attenuator setting had acceptable electron densities for between two heights, then the median of the acceptable electron densities was used. Using these three methods, the computer results were combined to give one electron-density profile for every 4 attenuator settings. Using either average or medians, electron densities of different heights or of different times were combined as discussed in Section 4.2.

4.2 *Electron-Density Results*

The results are presented in two forms: by the total differential absorption below each height (A_x/A_o ratios) and by electron densities. The A_x/A_o ratios given in Figures 4.2, 4.3, and 4.4 are plotted using a sixth order polynomial approximation of the ratio as calculated by method one. The eclipse shows a reduction in absorption which indicated a reduction in electron density as expected. The third day shows irregular absorption with a large increase in absorption. Referring to Figure 2.2. the increase in absorption is related to the *X*-ray flux burst. The electron density for above 75 km for the three days given in Figure 4.5 shows a good correlation between the large increase in electron density on July 11 and the burst of *X*-ray flux. Due to this obvious contamination, the second control day is not used for comparison during the burst period.

Figure 4.6 gives the A_x/A_o ratios versus height. The ratios were determined using method one and taking the median of the groups within the hour corresponding to the maximum obscuration of the solar eclipse (1400-1500 CST). Due to the much larger absorption in the control days than during the eclipse, the electron densities above 81 km (approximately) are not valid according to method two and three, but with the eclipse day, the values should be acceptable up to 85 km.

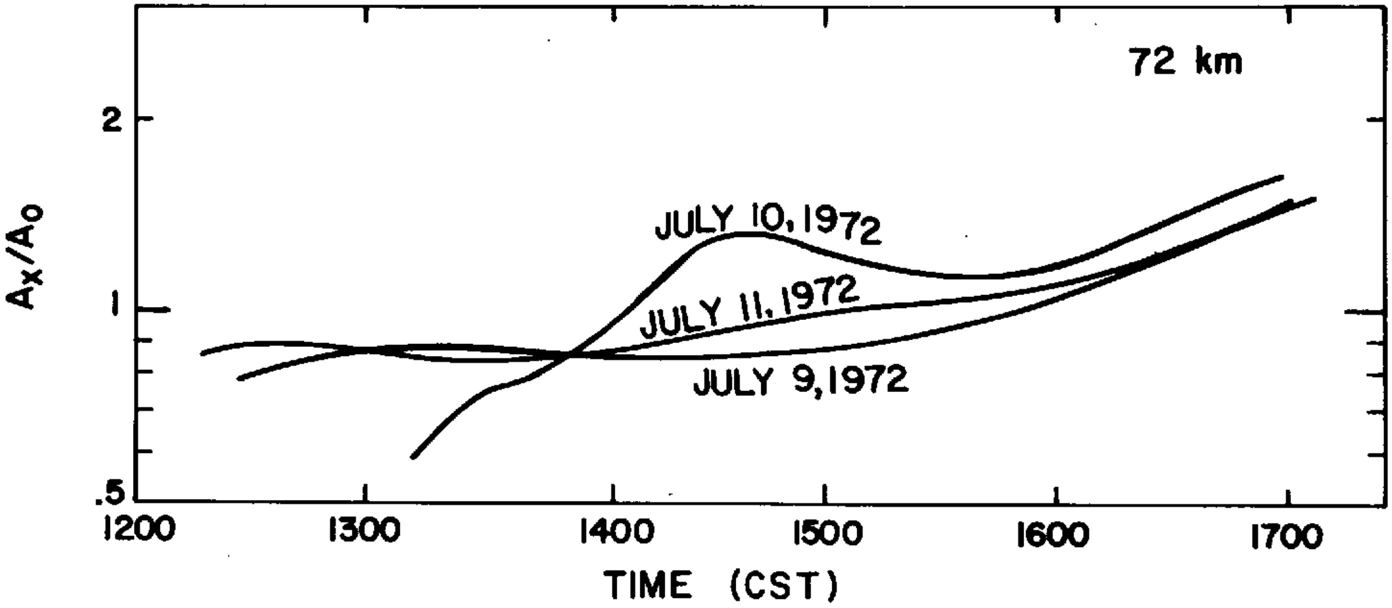

Figure 4.2 Comparison of the A_x/A_o ratio at 72 km for July 9, 10, and 11.

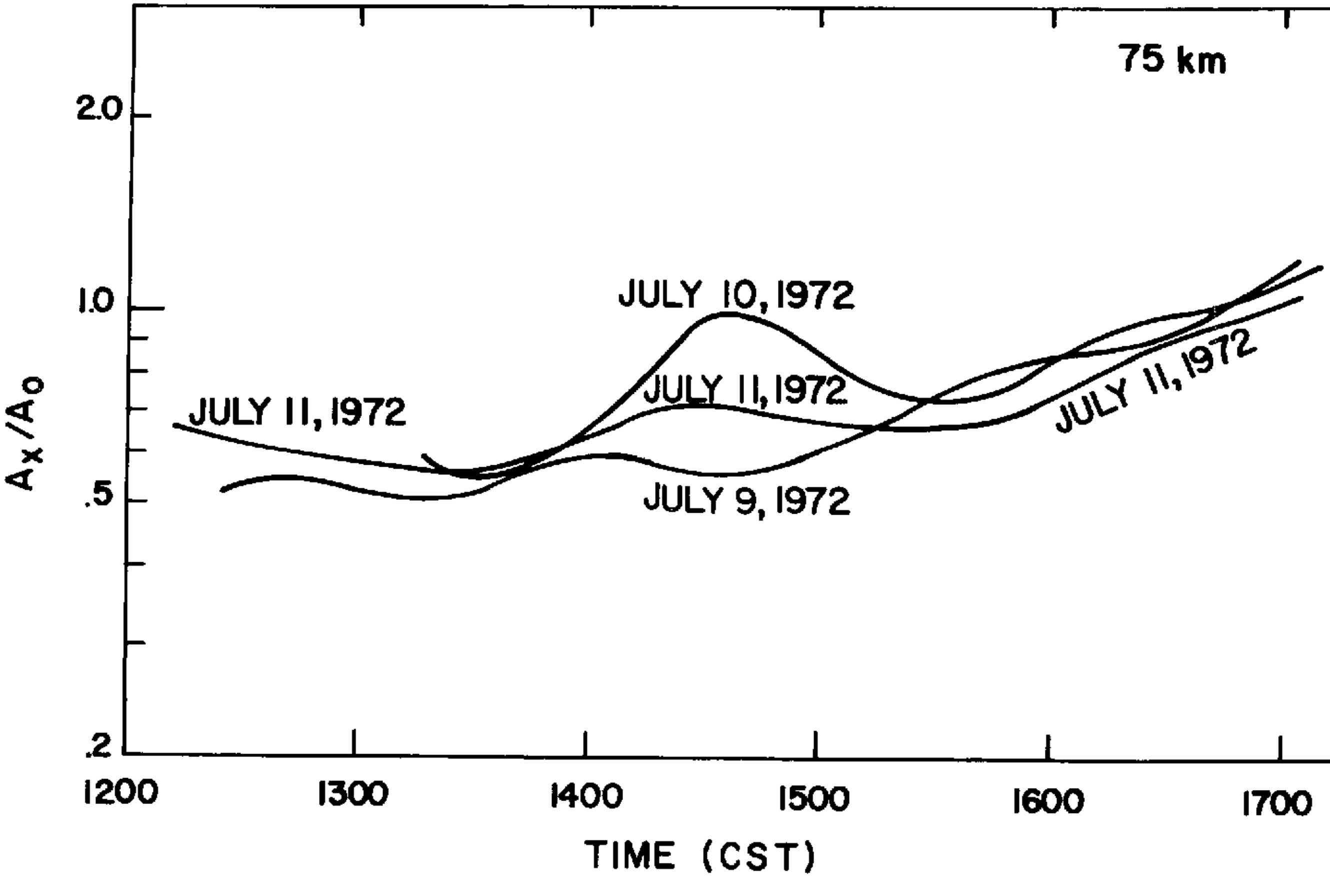

Figure 4.3 Comparison of the A_x/A_o ratio at 75 km for July 9, 10, and 11.

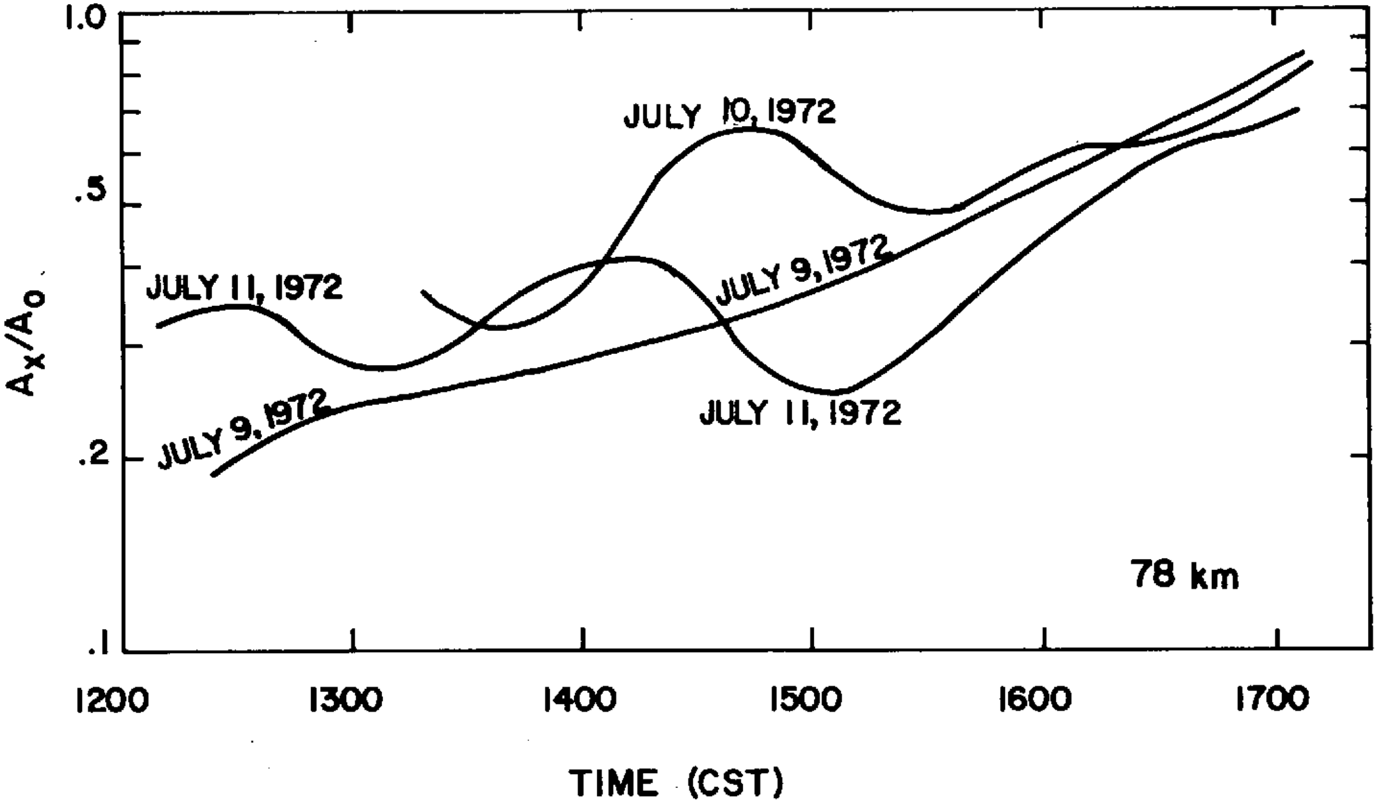

Figure 4.4 Comparison of the A_x/A_o ratio at 78 km for July 9, 10, and 11.

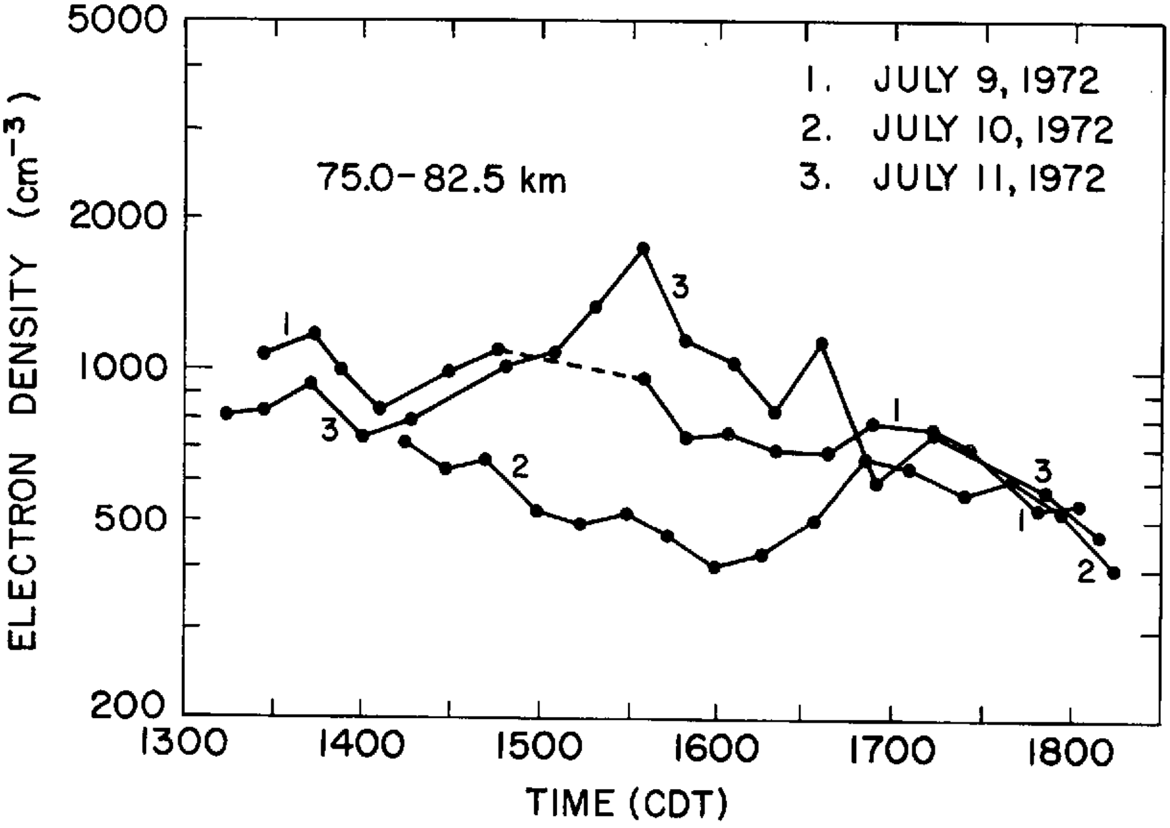

Figure 4.5 Median electron densities between 75 and 82.5 km.

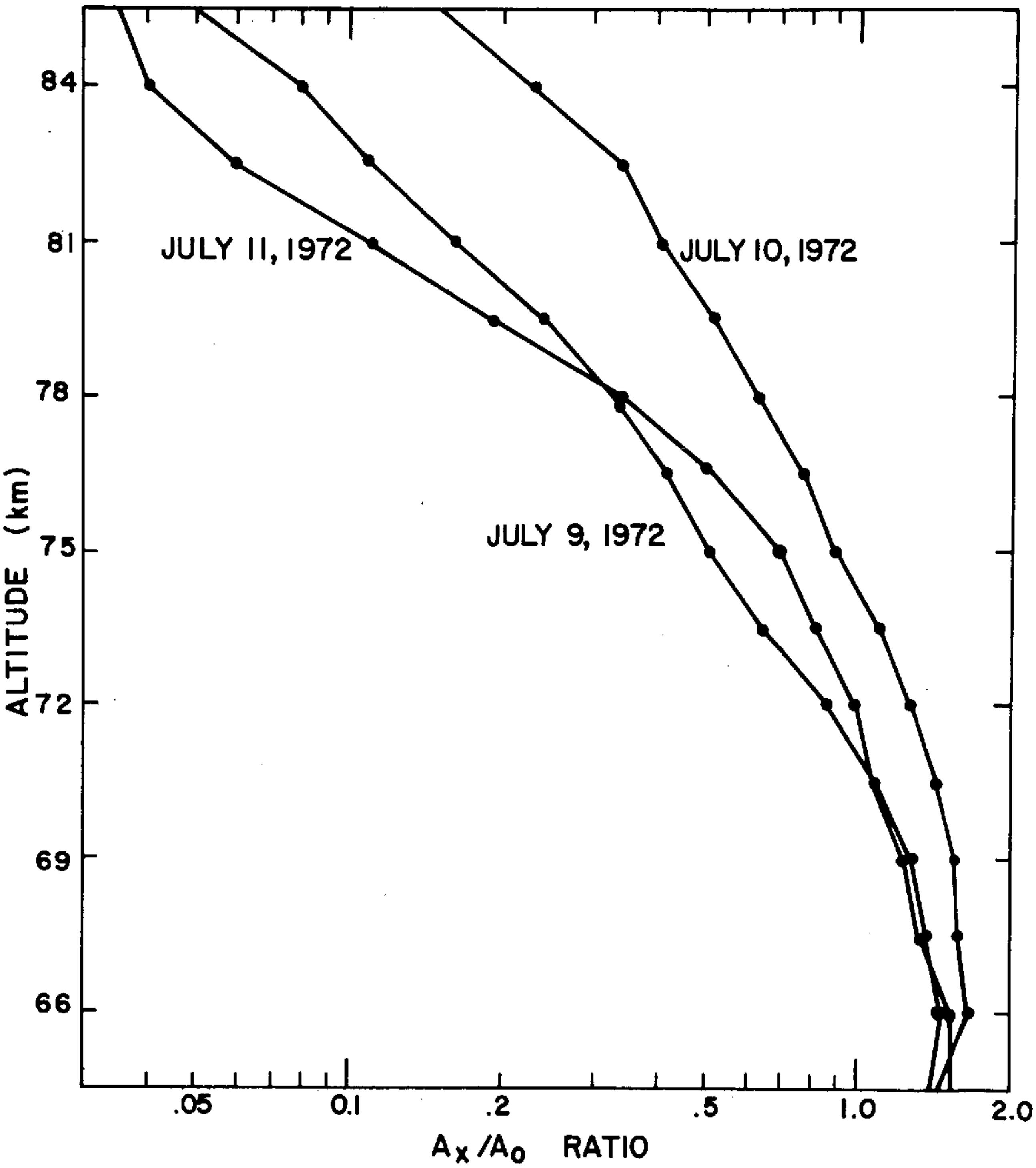

Figure 4.6 Median A_x/A_o profiles between 1400 and 1500 CST for each day.

Figure 4.7 gives the electron-density variation with time. The electron densities are averages between 70.5 and 78 km and between 78 and 87.5 km with the electron densities obtained by using method one for processing the computer result. Figure 4.5 and 4.8 give the electron-density median for 75 to 82.5 km and 67.5 and 75 km, respectively, as each varies with time. These electron densities were obtained using method three. At the lower altitudes, the median electron densities show no effect from the eclipse while the average electron densities do show a slight effect. This difference, though, is mainly attributed to the higher heights the averages were taken from rather than to the method used. The highest heights show large effects due to the eclipse. Figure 4.7 shows a minimum electron density near maximum obscuration of the eclipse while Figure 4.5 shows the minimum being delayed by half an hour. This is attributed to the variation in the data due to the inaccuracies in the partial-reflection equipment. The X-ray burst shown in Figure 2.2 seems to have no effect at the lower altitudes.

Median electron-density variations with height are given in Figure 4.9. These values are the median obtained by processing the computer results utilizing method two and finding the median value between 1400 and 1500 CST. Below 75 km the eclipse does not seem to have much effect on the electron density as shown in Figure 4.9, but above 75 km, the electron density decreases by 45 to 65%. The upper height for this comparison is 81 km due to the small A_x/A_o ratios (shown in Figure 4.6). The electron-density profile shows some conformity to the expectation given in Section 2.4.

4.3 *Theoretical Applications*

Since the eclipse never reached totality, the electron production (q) cannot be assumed to be zero, but equation (2.4) can be used as an approximation

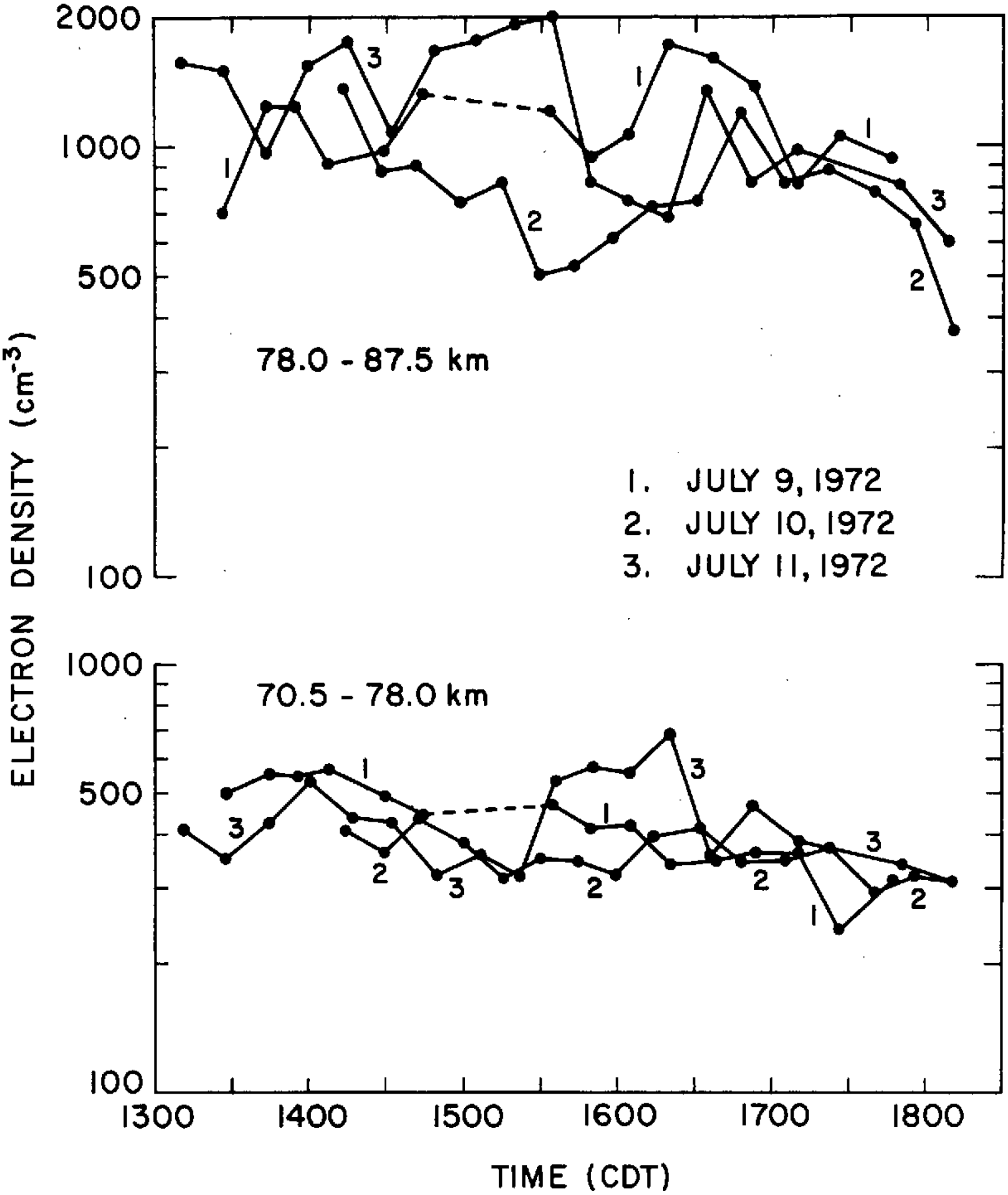

Figure 4.7 Average electron densities between the altitudes 78.0 - 82.5 km and 70.5 - 78.0 km.

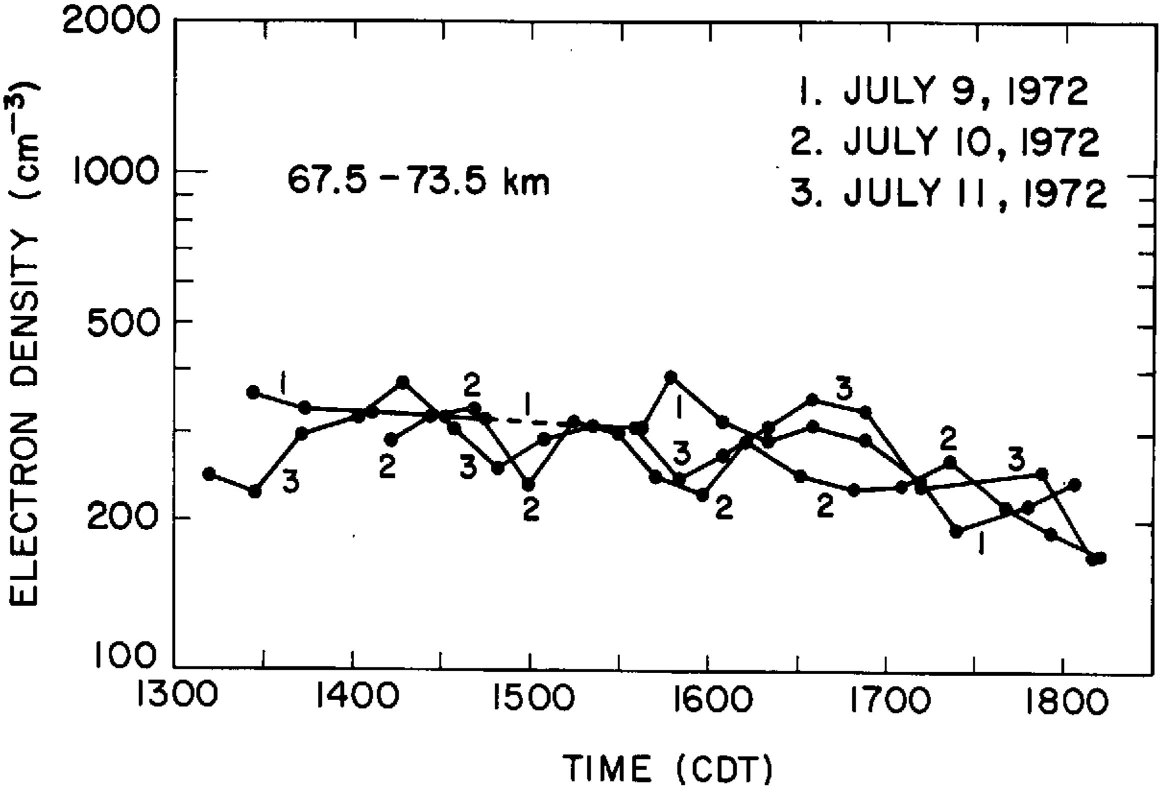

Figure 4.8 Median electron densities between 67.5 and 75 km.

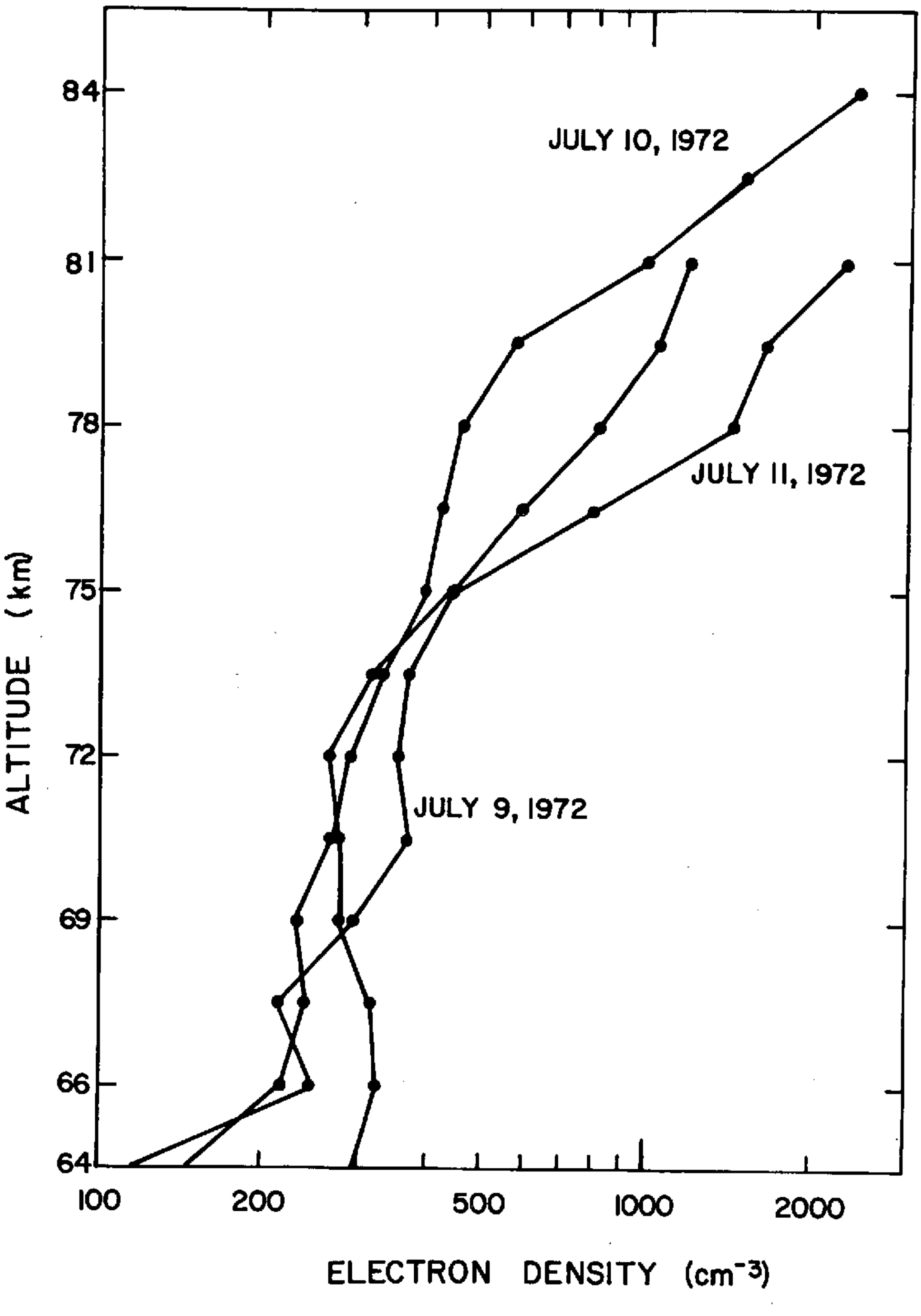

Figure 4.9 Median electron-density profiles between 1400 and 1500 CST.

to $[e]$ and α_{eff}. Equation (3.1)

$$q = \sigma_i(\mathrm{NO})\ [\mathrm{NO}]\ I_\infty\ e^{-\tau}\ F_o \qquad (3.1)$$

where q = electron production rate in $\mathrm{cm}^{-3}\ \mathrm{sec}^{-1}$

$\sigma_i(\mathrm{NO})$ = ionization cross-section of nitric oxide = $2 \times 10^{-18}\ \mathrm{cm}^2$

$[\mathrm{NO}]$ = number density of nitric oxide in cm^{-3}

I_∞ = incident Lyman-alpha flux at the top of the atmosphere = 3.1×10^{11} photons $\mathrm{cm}^{-2}\ \mathrm{sec}^{-1}$

F_o = the function of the unobscured solar disk

τ = optical depth

given by *Sechrist* [1966], was used to approximate the electron production rate and equation (3.2) was used to approximate the optical depth.

$$\tau = \sigma_a(\mathrm{O}_2)\ [\mathrm{O}_2]\ H \sec\chi \qquad (3.2)$$

$[\mathrm{O}_2]$ = number density of molecular oxygen in cm^{-3}

H = scale height

χ = solar zenith angle

Figure 4.10 shows the variation of q during the eclipse as compared to the variation without the eclipse. The electron production rates were used to obtain theoretical electron densities with α_{eff} being chosen to give the best fit to the experimental results. A value of 2×10^{-6} for α_{eff} was determined for the eclipse day between 75 and 82.5 km and 1.77×10^{-6} for the same height range on the control days. For the heights 78 to 87.5 km α_{eff} was found to be 8.46×10^{-7}. These values for α_{eff} are similar to ones given by *Mitra* [1968]. Figure 4.11 shows a comparison between the theoretical $[e]$ during the eclipse and without the eclipse using α_{eff} of 1.77×10^{-6}.

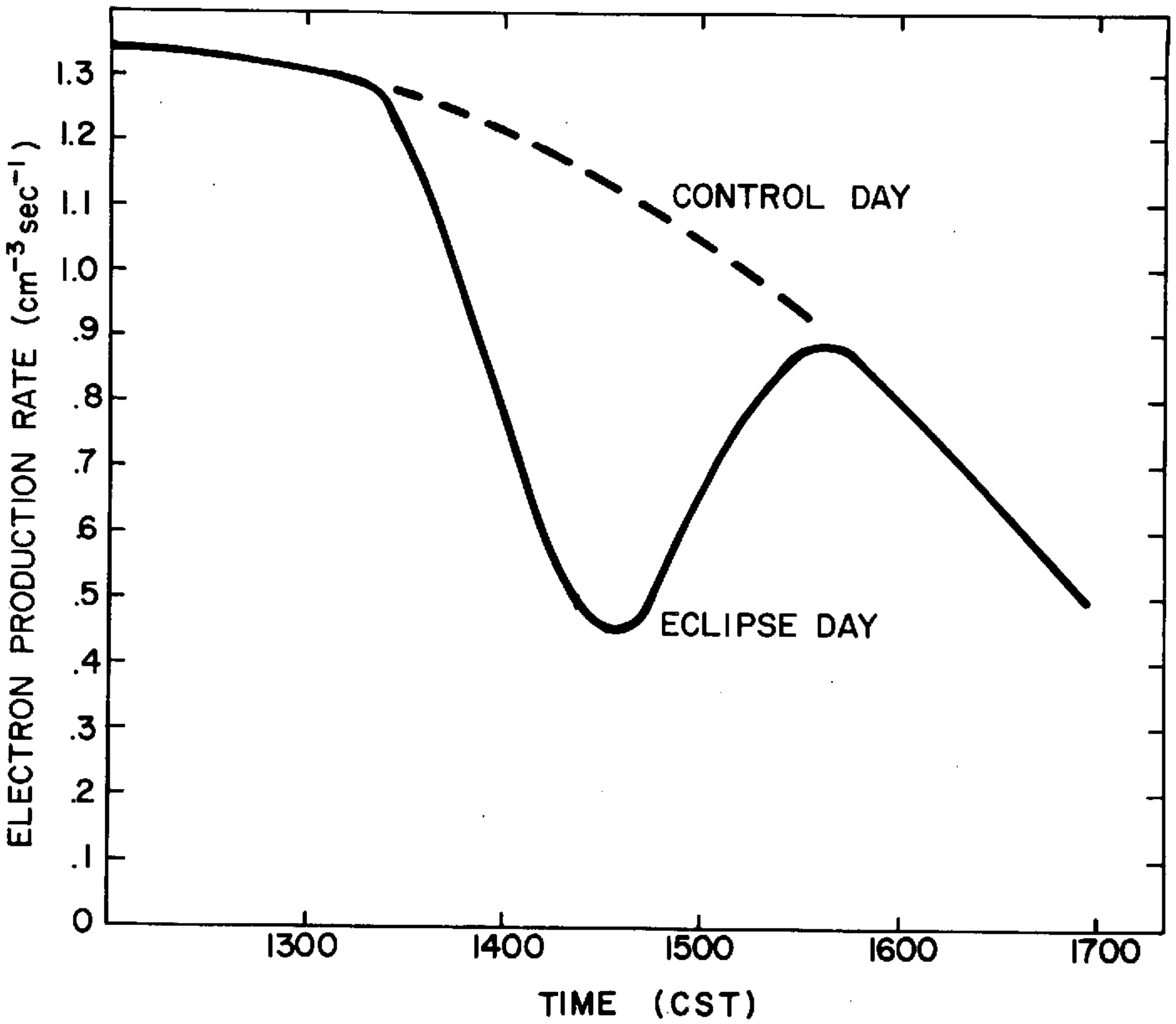

Figure 4.10 Electron production rate between 75 and 82.5 km during the eclipse and during the control days. The NO distribution used is from *Meira* [1971].

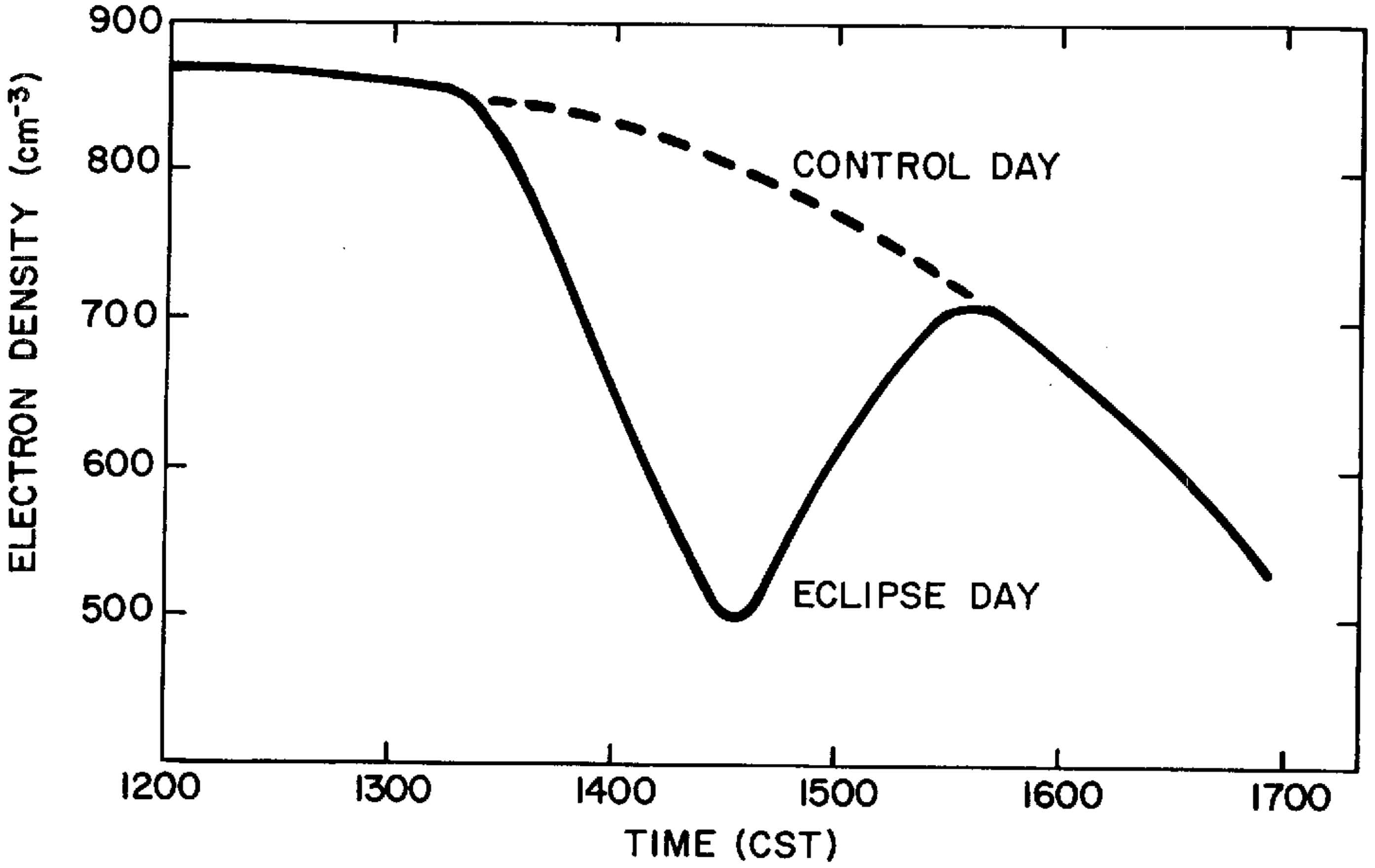

Figure 4.11 Theoretical electron densities between 75 and 82.5 km for eclipse and control day; calculated using an α_{eff} of 1.77×10^{-6}.

The electron density of the eclipse was divided by average electron density of the control data and compared to the obscuration function as seen in Figure 4.12. The comparison of the experimental [*e*] during the eclipse and the theoretical [*e*] without the eclipse using equation (2.4) was also made and is shown in Figure 4.13.

The electron density for July 9 shows a good correlation with the solar zenith angle (Figure 4.14) and was therefore divided by the theoretical [*e*] to eliminate the effects of the solar zenith angle and to determine the variability of the experimental [*e*] (Figure 4.15). The same comparison is made with the eclipse [*e*] (Figure 4.15) and shows a similar but greater variability.

Generally, the eclipse electron densities show a decrease that is greater than expected from the equation (2.4). Other than the possibility that this is caused by variabilities due to inaccuracies in the experiment, there are three reasons why this may occur:

1. The obscuration function of the ionization source (Lyman-α) is different than the uniform-disk obscuration function used.
2. The α_{eff} increased during the eclipse. This could be caused by a change in the hydrated-ion composition between 75 and 81 km.
3. Loss by attachment is increased by the eclipse.

The electron-density profiles in Figure 4.10 show good comparison with the profile with 40% obscuration given in Figure 2.7 and with 60% obscuration shown in Figure 2.6. *Smith, et al.* [1965] described small changes below 70 km as the *C*-layer caused by cosmic rays which disappear as the eclipse reaches totality. The effect can be seen up to 69 km in Figure 4.9.

4.4 *Summary*

Comparing Figure 4.9 Figures 2.6 and 2.7, the electron-density profiles of this eclipse are similar to previous eclipses for the same obscuration. Generally, similar conclusions can be drawn. The difficulty in interpreting the

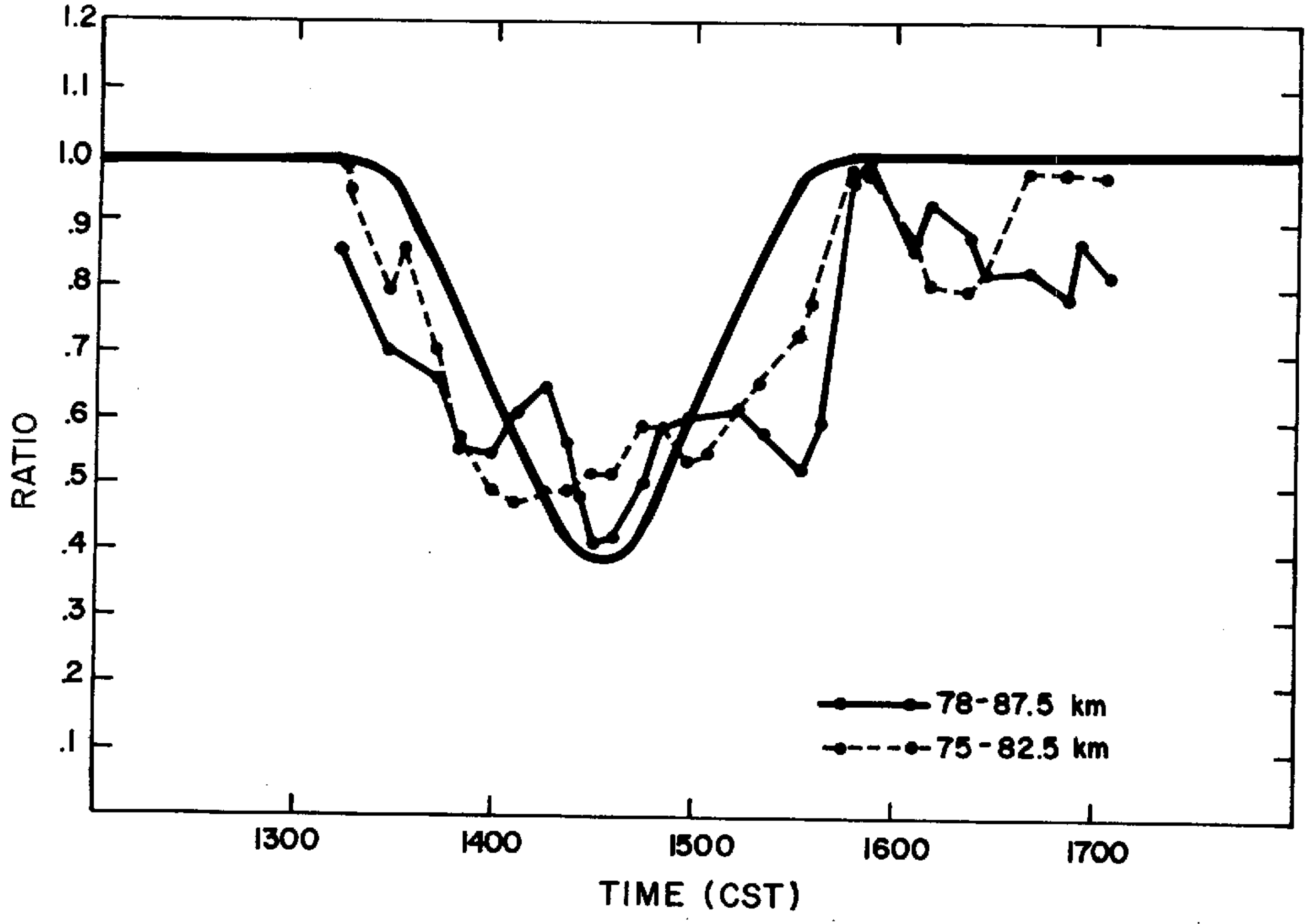

Figure 4.12 The ratio of electron densities for the average of the control day as compared to the unobscured sun.

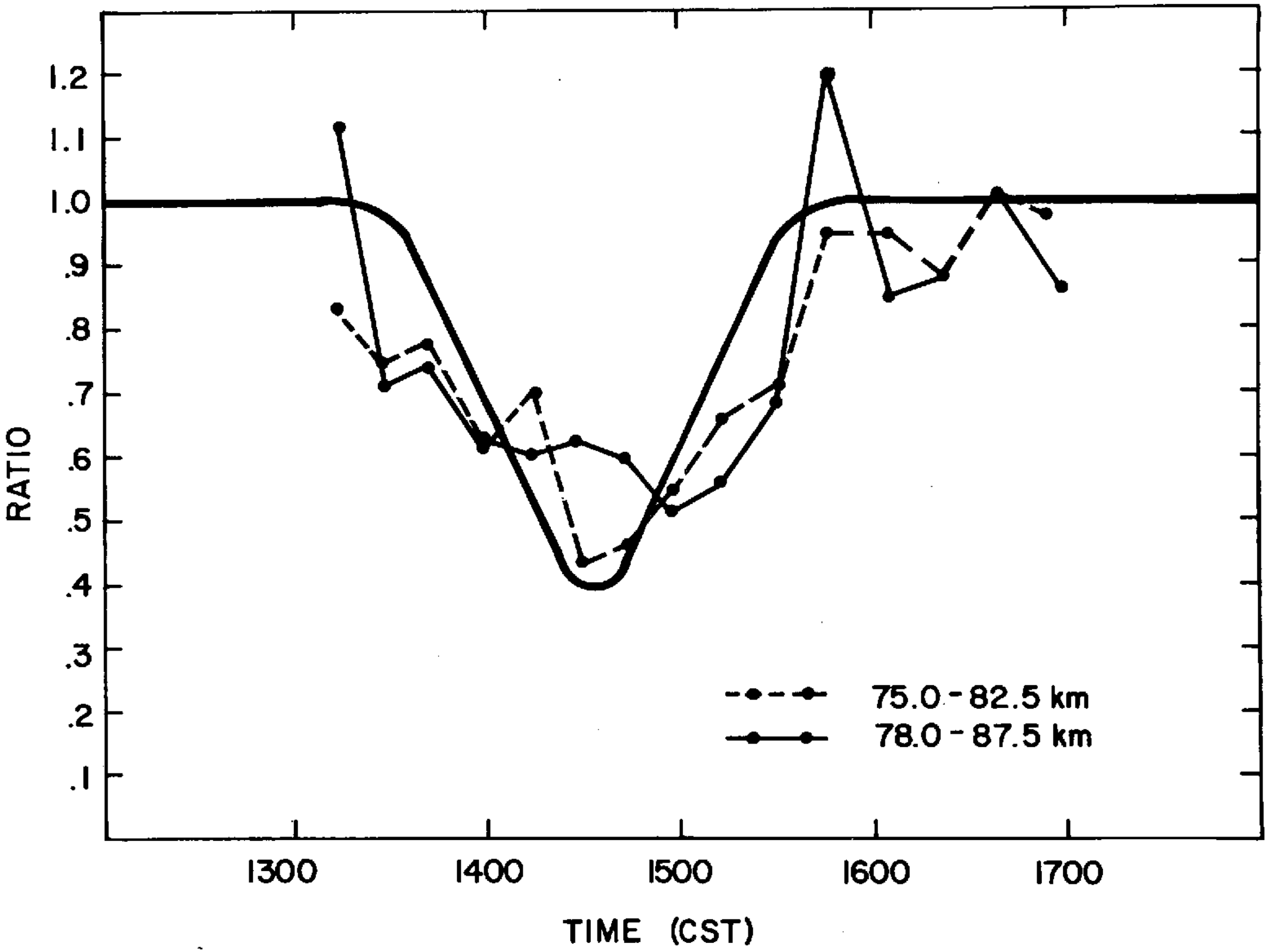

Figure 4.13 The graph of the ratios of the theoretical [*e*] for the unobscured sun to the experimental [*e*] for the eclipse as compared to the unobscured sun. The α_{eff} used for 75 to 82.5 km is 1.77×10^{-6} and for 78 to 87.5 km is 8.46×10^{-7}.

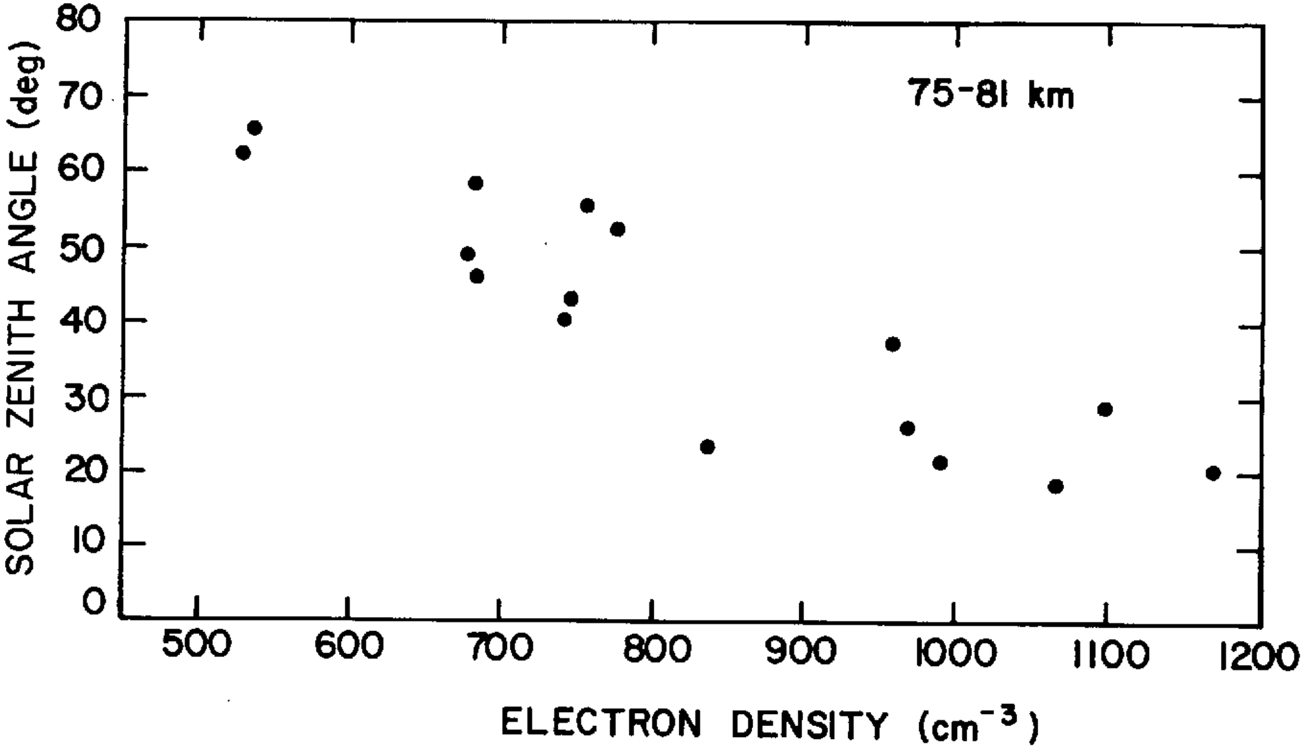

Figure 4.14 Scatter plot correlating the electron density for July 9, 1972 between 75 and 82.5 km to the solar zenith angle.

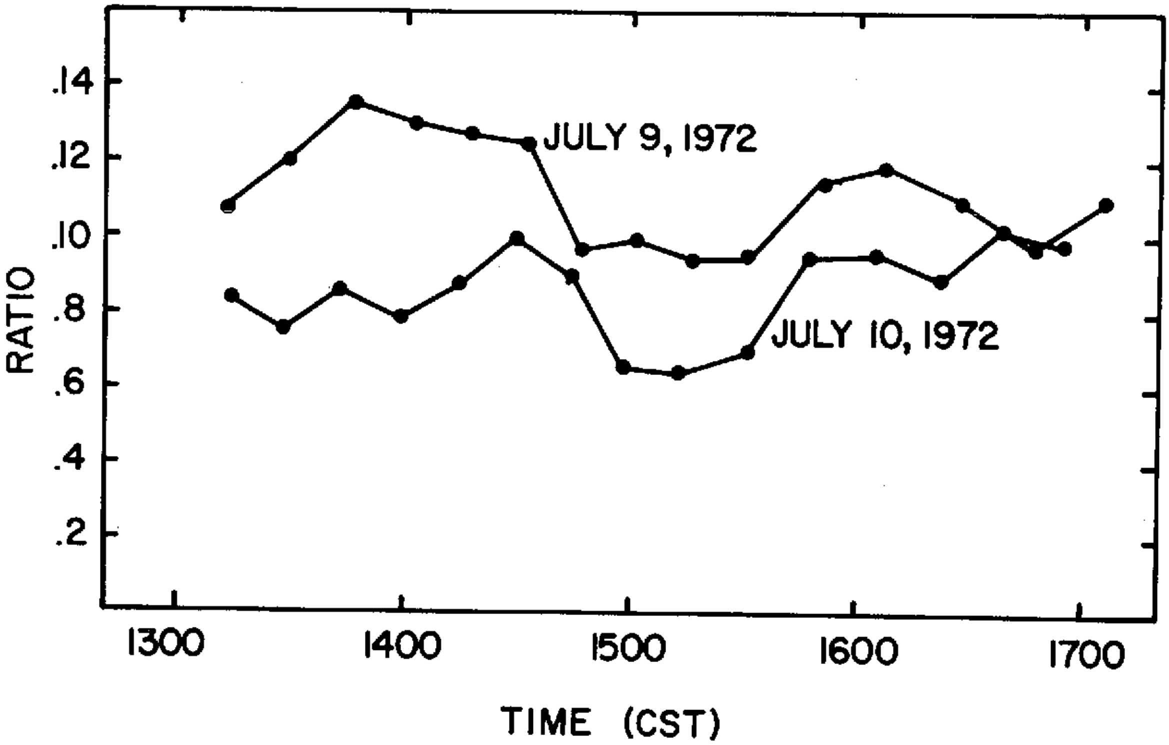

Figure 4.15 The graph of the ratio of theoretical electron densities to the experimental electron densities for July 9, 10, 1972.

results lies in the variation of the electron density of the eclipse with time. In Figures 4.12 and 4.13 a small decrease in electron density precedes the obscuration of the sun. An error of 20% can be expected due to the equipment and 20% error can be expected in the collision frequencies. Errors due to collision frequencies will cancel in Figure 4.12 but the errors due to the equipment will increase. For Figure 4.13, the reverse is true, but there are also errors due to the approximations made in equations (2.4), (3.1), and (3.2). With these possibilities of errors and observing that the ratio after the eclipse can get as low as .8 in Figure 4.12, the initial decrease can be interpreted as experimental error. The errors in Figure 4.13 can be seen in the variations in Figure 4.15.

No correlation could be seen between the X-ray flux and the electron density on the third day except furing the X-ray burst period. Therefore, Lyman-α is assumed to be the main ionization source and the theoretical calculations were made on that assumption.

Of the three reasons for the large decrease in $[e]$, the effects due to changes in hydrated ions is the most likely. During the day electron loss by attachment is insignificant above 75 km. Since the obscuration of the sun was only 60% which corresponded to a production rate similar to that of 65° solar zenith angle, the loss process would still be by recombination.

The larger concentrations of Lyman-α on the solar disk were in the southern hemisphere and were not obscured and the intensity of 1-8 Å X-ray flux was too small to have any large effect. Therefore, the obscuration function of the ionizing source would have the same obscuration or less. This leaves the only possibility for the larger decrease in free electron as being due to changes in the hydrated ions.

5. CONCLUSIONS

The solar eclipse provides a good opportunity to study several processes of the *D* region and to develop its theoretical model. Accurate interpretation of the eclipse data is required to determine exactly the *D*-region ion production and loss processes, the variation of α_{eff}, formation of hydrated ions, and negative ion chemistry. A brief theory of the *D*-region chemistry is presented in Chapter 2 and used to analyze the data in Chapter 4. The equipment used in the collecting and processing of the partially reflected waves, as well as the refinements made in the collection process are given in Chapter 3. The newer partial-reflection system, discussed in Chapter 3, has been in use for the daily collection of data. Results from this newer system are given by *Denny and Bowhill* [1973]. This chapter reviews the results of the partial-reflection data taken during the eclipse and suggests further developments of the partial-reflection system.

5.1 *Review of Results*

The effect of the eclipse below 75 km is below the experimental errors. These errors are due to the variability of receiver gain caused by temperature fluctuations, the 40 μsec pulse width of the transmitter, inaccuracies in the collision frequencies, and inaccuracies in noise reduction. In comparing the [*e*] profiles for July 9 and 10, 1972 in Figure 4.8, the beginning of the formation of a *C* layer can be seen resulting from cosmic rays. From Section 2.1, the main ionization source between 70 and 80 km is Lyman-α since the X-ray source effects were not observed below 81 km except when the X-ray flux increased above 1×10^{-3} erg cm^{-2} sec^{-1}.

The decrease between the electron density from July 9 and from July 10 is dependent on the height and is very marked between 79 and 81 km. Near 80 km, this change in electron density is as much as 55% between the results of July 9 and 10, which was not expected according to equation (2.4). The most probably answer given in Chapter 4 is that it is due to an initial large decrease in hydrated positive ions which are the major ions between 75 and 80 km during the daytime (as seen in Figure 5.1 by *Krankowsky, et al.* [1972]).

The theoretical [*e*] were used to compare with the experimental [*e*] in Figure 4.13 to remove any electron density variability not due to the eclipse. The results in Figures 4.12 and 4.13 show unexpected initial decreases in [*e*] prior to the eclipse and larger decreases than would be expected during the eclipse, but allowing for 20% error in these results, these variations are within the error limits. In general, there is good agreement with the data from *Smith, et al.*, [1927] and *Deeks* [1966].

5.2 *Suggestions for Further Work*

The present partial-reflection system has proved invaluable in presenting variations in electron densities diurnally and from day-to-day as presented by *Denny and Bowhill* [1973]. The system has several limitations, though. Either the signal-to-noise-ratio should be increased or the rates of data collection increased. Both of these changes would require alterations in the transmitter. By doubling the peak power of the transmitter, meaningful partial reflections could be obtained at lower altitudes without excessively disturbing the ionosphere due to the slow pulse rate as is done in the cross modulation experiment. By increasing the pulse rate, more data could be collected in the same interval of time, allowing for a more accurate statistical evaluation of the noise.

A new receiver has been built as mentioned in Chapter 3. The initial results obtained using it show an improvement in the results, but the problem of eliminating atmospheric noise remains. The main problem lies in defining the noise.

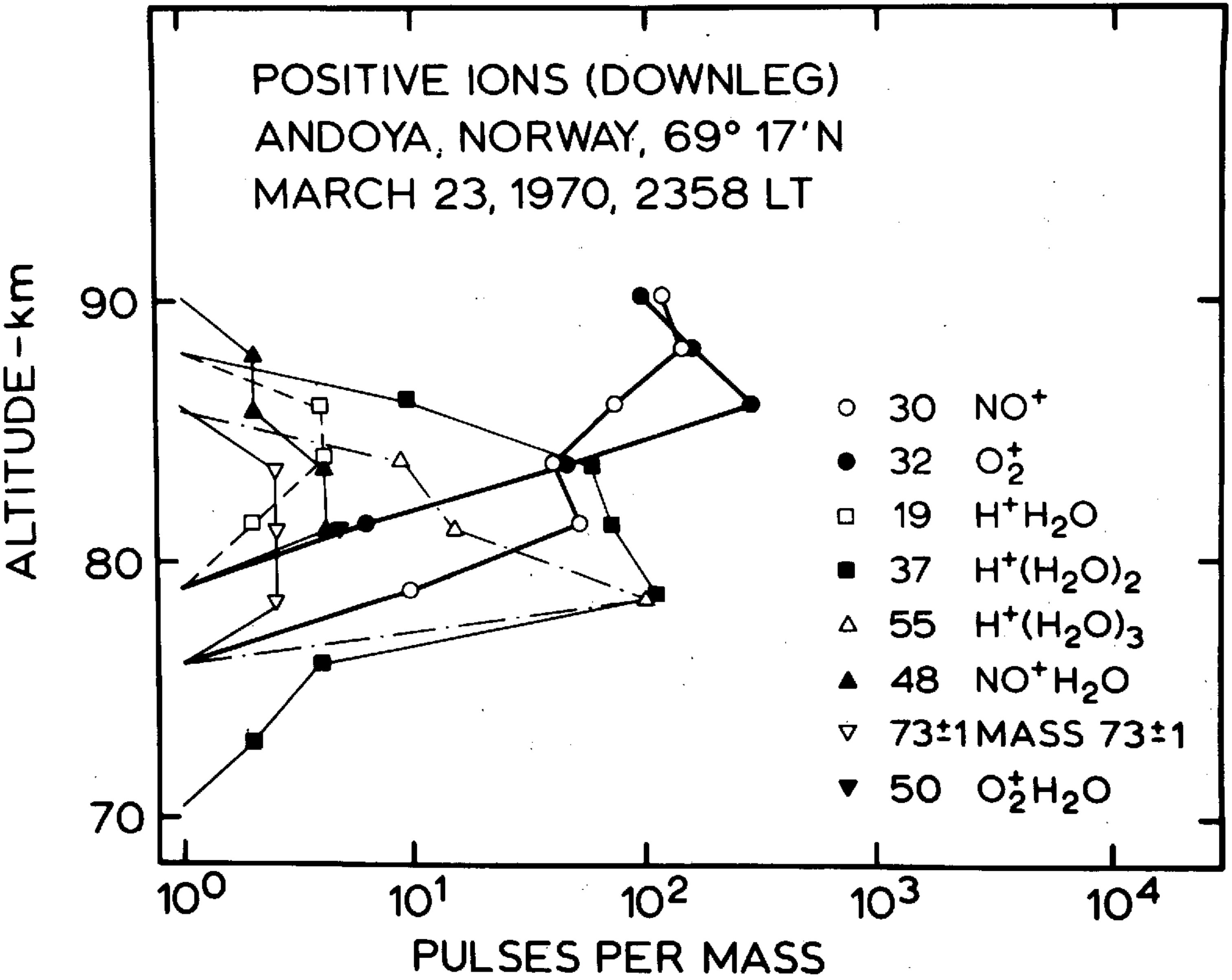

Figure 5.1 Rocket measurements of the positive-ion chemistry by *Krankowsky, et al.* [1972].

A study should be done on the specific types of noise received and the algorithms required to reject each. This would include receiving and storing noise on DECtape for later evaluation of the amplitude and phase.

A digital input/output would increase the efficiency of the collection and process. Presently the system requires the assistance of the operator every 3-1/2 minutes and uses one page of computer paper for every page of data. With a digital input/output, the computer could set the attenuators and control other switching which would free the operator for other tasks. This would also improve the usefulness of taking differential phase measurements as described by *Wiersma and Sechrist* [1972].

Using a line printer for outputting the data would allow for more sophisticated and complicated processing of data. This would also be required if the rate of collection is increased. To collect one file of data takes 3.5 minutes, to process one, about 45 sec, but to print out the results on the teletype and paper tape takes 2.6 minutes. Therefore, the processing would not be able to keep up with a faster collection unless the speed of printing the results increased.

REFERENCES

Aikin, A. C. (1972), The relationship of theory and experiment in the *D* region, *J. Atmos. Terr. Phys. 34*, 1591-1599.

Barth, C. A. (1966), Rocket measurement of nitric oxide in the upper atmosphere, *Planet. Space Sci. 14*, 623-630.

Belrose, J. S. and M. J. Burke (1964), Study of the lower ionosphere using partial reflection, *J. Geophys. Res. 69*, 2799-2818.

Birley, M. H. and C. F. Sechrist, Jr. (1971), Partial-reflection data collection and processing using a small computer, *Aeron. Rep. No. 42*, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

COSPAR International Reference Atmosphere (1965), North Holland Publishing Company, Amsterdam.

Deeks, D. G. (1966), *D*-region electron distributions in middle latitudes deduced from the reflexion of long radio waves, *Proc. Roy. Soc. A291*, 413-437.

Denny, B. W. and S. A. Bowhill (1973), *D*-region electron densities for the winter of 1972-73, *Aeron. Rep. No. 56*, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign

Donahue, T. M. (1972), Positive ion chemistry of the *D* and *E* regions, *Radio Sci. 7*, 73-80.

Edwards, B. (1972), Research in Aeronomy: October 1, 1972 - March 31, 1973, *Prog. Rep. 73-1*, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Fehsenfeld, F. C., A. L. Schmeltekopf, and E. E. Ferguson (1965), Correction in the laboratory measurement of the rate constant for $N_2^+ + O_2 \rightarrow N_2 + O_2^+$ at 300° K *Planet. Space Sci. 13*, 919-920.

Fehsenfeld, F. C. and E. E. Ferguson (1969), Origin of water cluster ions in the *D*-region, *J. Geophys. Res. 74*, 2217-2225.

Fejer, J. A. and R. W. Vice (1959), An investigation of the ionospheric *D* region, *J. atmos. Terr. Phys. 16*, 291-306.

Ferguson, E. E. (1971), *D*-region ion chemistry, *Rev. Geophys. Space Phys. 9*, 997-1008.

Gardner, F. F. and J. L. Pawsey (1953), Study of the ionospheric *D*-region using partial reflections, *J. Atmos. Terr. Phys. 3*, 321-344.

Good, A., D. A. Durden, and P. Kebarle (1970), Ion-molecule reactions in pure nitrogen and nitrogen containing traces of water at total pressures .5-4 torr. kinetics of clustering reactions forming $H^+(H_2O)_n$, *J. Chem. Phys. 52*, 222-229.

Gregory, J. B. (1956), Ionospheric reflections from heights below the *E* region *Australian J. Phys. 9*, 324-342.

Henry, G. W., Jr. (1966), Instrumentation and preliminary results from shipboard measurements of vertical incidence ionospheric absorption, *Aeron. Rep. No. 13*, Aeron. Lab., Dep. Elec. Eng. Univ. Ill., Urbana-Champaign.

Huffman, R. E., D. E. Paulsen, J. C. Larrabee, and R. B. Cairns (1971), Decrease in *D*-region $O_2(^1\Delta_g)$ photoionization rates resulting from CO_2 absorption, *J. Geophys. Res. 76*, 1028-1038.

Hunt, B. G. (1965), A theoretical study of the changes occuring in the ozonosphere during a total eclipse of the sun, *Tellus XVII*, 516-523.

Hunten, D. M. and M. B. McElroy (1968), Metastable $O_2(^1\Delta)$ as a major source of ions in the *D* region, *J. Geophys. Res. 73*, 2421-2428.

Keneshea T. J. (1967), A technique for solving the general reaction-rate equations in the atmosphere, *Environmental Res. Papers No. 263*, Air Force Cambridge Research Laboratories, Bedford, Massachusetts.

Krankowsky, D., F. Arnold, H. Wieder, J. Kissel, and J. Zähringer, (1972), Positive-ion composition in the lower ionosphere, *Radio Sci. 7*, 93-98.

Lauter, E. A., and R. Knuth (1967), Precipitation of high energy particles into the upper atmosphere at medium latitudes after magnetic storms, *J. Atmos. Terr. Phys. 29*, 411-417.

Lodato and Mechtly (1971), Rocket measurements of electron collision frequency, *Aeron. Rep. No. 45*, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Mechtly, E. A., S. A. Bowhill, L. G. Smith, and H. W. Knoebel, (1967), Lower ionosphere electron concentrations and collision frequency from rocket measurements of Faraday rotation, differential absorption, and probe current, *J. Geophys. Res. 72*, 5239-5245.

Mechtly, E. A., C. F. Sechrist, Jr., and L. G. Smith (1972), Electron loss coefficient 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., Jr. (1971), Rocket measurements of the upper atmospheric nitric oxide and their consequences to the lower ionosphere, *J. Geophys. Res. 76*, 202-212.

Mitra, A. P. (1966), An ionospheric estimate of nitric oxide concentration in the *D*-region, *J. Atmos. Terr. Phys. 28*, 945-955.

Mitra, A. P. (1968), *D*-region processes in non-polar latitude, *J. Atmos. Terr. Phys.* 30, 1065-1114.

Mitra, A. P. and J. N. Rowe (1972), Ionospheric effects of solar flares - VI changes in *D*-region ion chemistry during solar flares, *J. Atmos. Terr. Phys. 34*, 795-806.

Narcisi, R. S. and A. D. Bailey (1965), Mass spectrometric measurements of the positive ions at altitudes from 64 to 112 kilometers, *J. Geophys. Res. 70*, 3687-3700.

Pearce, J. B. (1969), Rocket measurements of nitric oxide between 60 and 96 kilometers, *J. Geophys. Res. 74*, 853-861.

Pirnat, C. R. and S. A. Bowhill (1968), Electron densities in the lower ionosphere deduced from partial reflection measurements, *Aeron. Rep. No. 29*, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Radicella, S. M. and D. W. Stowe (1970), *D*-region ion chemistry, *Aeron. Rep. No. 38*, Aeron. Lab., Dep. Elec. Eng. Univ. Ill., Urbana-Champaign.

Reynolds, D. A. and C. F. Sechrist, Jr. (1970), Measurement of average electron density between 75 and 80 kilometers, *Aeron. Rep. No. 36*, Aeron. Lab., Dept. 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., Calif.

Sechrist, C. F., Jr. (1966), A theory of the winter absorption anomaly at middle latitudes, *J. Atmos. Terr. Phys. 29*, 113-136.

Sechrist, C. F., Jr. (1972), Theoretical models of the *D*-region, *J. Atmos. Terr. Phys. 34*, 1565-1589.

Sen, H. K. and A. A. Wyller (1960), On the generalization of the Appleton-Hartree magnetoionic formulas, *J. Geophys. Res. 65*, 3931-3950.

Shimazaki, T. and A. R. Laird (1970), A model calculation of the diurnal variation in minor neutral constituents in the mesosphere and lower thermosphere including transport effects, *J. Geophys. Res. 75*, 3221-3235.

Shimazaki, T. and A. R. Laird (1972), Seasonal effects on distributions of minor neutral constituents in the mesosphere and lower thermosphere, *Radio Sci. 7*, 23-44.

Smith, L. G., C. A. Accardo, L. H. Weeks, and P. J. McKinnon (1965), Measurements in the ionosphere during the solar eclipse of 20 July 1963, *J. Atmos. Terr. Phys. 27*, 803-829.

Somayajulu, Y. V. and M. B. Avadbanula (1972), Rocket measurements of *D*-region electron densities at the equator, *Indian J. Radio Space Sci. 1*, 81-83.

Thomas, L. (1971), The lower ionosphere, *J. Atmos. Terr. Phys. 33*, 157-195.

Turco, R. P. and C. F. Sechrist, Jr. (1970), An investigation of the ionospheric *D* region at sunrise, *Aeron. Rep. No. 41*, Aeron. Lab., Dep. Elec. Eng. Univ. Ill., Urbana-Champaign.

U. S. Standard Atmosphere Supplements (1966), U. S. Government Printing Office, Washington, D. C.

U. S. Department of Commerce (1972), *Solar-Geophysical Data* 337, part I, 58-59.

U. S. Department of Commerce (1973), *Solar-Geophysical Data 341*, part II, 48-49.

Whitten, R. C. and I. G. Poppoff (1971), *Fundamentals of Aeronomy*, John Wiley and Sons, Inc., New York.

Wiersma, D. J. and C. F. Sechrist, Jr. (1972), Differential phase measurements of *D*-region partial reflections, *Aeron. Rep. No.* 47, Aeron. Lab. Dep. Elec. Eng. Univ. Ill., Urbana-Champaign.

APPENDIX

```
        .TITLE  DLOGFI
/       DLOGFI IS A COMBINATION OF ALL THE PARTIAL REFLECTION
/ MACRO PROGRAMS USED IN COLLECTING AND PROCESSING DATA.  THE
/ PROGRAMS CONTAINED IN THIS VERSION ARE:
/INITIALIZATION:        INTIM           STARTS THE TIME
/                       DLOGF           INITAILIZES COLL.& PROC. PARAM.
/                       READM           READS UNFORMATTED CHAR. FROM TTY
/                       TOD             INCREMENTS THE TIME OF DAY
/CALIBRATION:           RADC            READS SAMPLES FROM A/D CONVERTER
/                               FOR FORTRAN PROGRAMS
/                       TTM             WRITES LIN. TABLE OUT ON DISK
/COLLECTION:            BEGIN           MAIN COLL. PROG.--A REAL TIME
/                               SUBROUTINE API LEVEL 6
/                       WT1             GIVES TIME TO LOWER API LEVELS
/                       RSUB            SETS UP DATA PACK & CHECKS NOISE
/                               --R.-T. SUB. AT API LEVEL 5
/                       PAC             PACKS DATA DOUBLE
/                       DTRANS          WRITES DATA ON STORAGE DEVICE
/                       CKCNT           CHECKS FOR ENOUGH DATA
/                       CHECK           SEES IF AX & AO IS IN THE RIGHT
/                                ORDER AND ARE COLL. IN PAIRS--API 5
/                       LIN             LIN. DATA,NEG #=0,& CHECKS NOISE
/                       ADREAD          PREPARES A/D CONVERTER READ
/                       ADINT           A/D INTERRUPT SERVICE ROUTINE
/PROCESSING:            CHNG            INIT. DEV. & CHECKS FOR COLL.FIE
/                       CONTR           WAITS FOR FILE TO BE COLLECTED
/                       WAITI           ALLOWS TIME FOR BACKGROUND
/                       CKCOL           CHECKS FOR UNWANTED COLL.STOP
/                       SWIT            USED TO SWITCH DISKS
/                       DUMPT           READS DATA, UNPACKS IT,& PUTS
/                               IT INTO A FORTRAN ARRAY
/
/       DATA IS COLLECTED ALTERNATELY ORDINARY AND EXTRAORDINARY
/AS DISCRIBED BY BIRLEY (AERONOMY REPORT 42).  THE TIMING OF THE
/DATA IS DETERMINED BY AN EXTERNAL ENCODE PULSE.  THE MODE OF THE
/DATA INPUTED INTO THE COMPUTER IS DETERMINED BY A TIMING PROGRAM
/(CHECK) SET UP BY D. WARD.  ESSENTIALLY HOW IT WORKS IS AFTER A
/FRAME OF DATA HAS BEEN READ IN THE COMPUTERS CLOCK IS SET FOR 9/60
/OF A SECOND (9 PULSES).  IF NO OTHER DATA IS READ IN BEFORE THE
/TIME EXPIRES THE DATA FRAME WAS EXTRAORDINARY MODE AND IS REJECTED.
/OTHERWISE BOTH FRAMES ARE ACCEPTED.  THIS CHECK IS MADE AT THE
/BEGINNING, AFTER EACH DATA TRANSFER, AND WHENEVER THE COLLECTION
/IS RESTARTED OR AN ERROR CONDITION EXIST.  THE PROGRAM IS SET SO
/AS TO NOT OVER MAXIMUM STORAGE ON THE DISK.
/
/""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
/
TTI=4                           /TELETYPE IN
TTO=6                           /TELETYPE OUT
TBI=10                          /.DAT SLOT OF LIN. TABLE
OUTPT=2                         /PLACE TO STORE DATA
OUTPT2=1                        /SECOND PLACE TO STORE DATA
DATIN=5                         /.DAT SLOT TO READ DATA
DATIN2=3                        /SECOND .DAT SLOT TO READ DATA
OUT=1                           /OUTPUT TO I/O DEVICE
IN=0                            /INPUT FROM I/O DEVICE
ASC=2                           /TYPE OF I/O MODE
IA=3                            /TYPE OF I/O MODE
DUMP=4                          /TYPE OF I/O MODE
SKAR=5                          /# OF DATA TO BE DELETED
TNSAM=37                        /# OF SAMPLES TO BE READ IN
NSAM=TNSAM-SKAR                 /# OF SAMPLES PER FRAME TO BE STORED
NSAMP=NSAM/2+1                  /SIZE OF ONE FRAME PACKED DOUBLE
DATBLK=TNSAM+2                  /SIZE OF INITIAL DATA BLOCK
DTBLK=374                       /SIZE OF 1 BLOCK OF STORAGE
DATSTR=DTBLK/NSAMP              /# OF FRAMES FOR 1 BLOCK OF STORAGE
RBLK=10                         /SIZE OF BLOCK FOR TTI READ
MNCC=-11                        /COUNT. FOR # OF NOISE FOR MAX. NOISE
NPFC=-6                         /MINUS (#+1) OF NOISE PER FRAME
NDPC=NSAM/2                     /# OF DATA PER FRAME
DPR=NDPC+1                      /RESET FOR POINTER TO DATA
MXFPD=350000/NSAMP              /MAX. # FRAMES PER DEVICE
NOFDK=1                         /# OF DISKS TO BE USED
        .GLOBL  CHNG,.DA,.AD,DUMPT,DLOGF,CKCOL,CONTL,PROC,RADC,TTM
        .GLOBL  TBFORL,PP7,VFPP
        .IODEV  1,2,3,4,5,6,10  /.DAT TO BE USED
/""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
/       THE FOLLOWING SUBROUTINE IS USED TO PREPARE THE
/COLLECTION AND PROCESSING PROGRAMS FOR MUNIPULATION OF DATA.
/THE VARIABLES OF THE MACRO PROGRAMS ARE STORED IN THIS SUBROUTINE.
/THEREFORE AFTER THIS SUBROUTINE IS EXECUTED IT IS WRITTEN OVER
/AND SHOULD NOT BE REENTERED (FOR IT WILL NOT EXIST).
/
```

```
/
DLOGF   0
TC2     JMS*    .DA             /GET SUBROUTINE PARAMETERS' ADDRESSES
SUM4    JMP     .+4             /JUMP AROUND PARAMETER LIST
DB      0                       /ADDR. OF THE ADDR. OF THE DB SETTINGS
DBC     0
TC3     0
RMSG1   LAC     MSG1+3          /SET UP RESET FOR TELETYPE
TC1     DAC     RMSG1           /   MESSAGE
DSTOR   JMP     ONC4            /GO TO NEXT EXECUTABLE STATEMENT
CTT1    LAW     NPFC            /COUNTS # OF NOISE PER FRAME
CTT2    LAW     MNCC            /COUNTS GROUPS OF NOISE FOR MAX. NOI.
CTT3    LAW     -2              /USED TO SWITCH STORAGE DEVICES (IF
CTT4    LAW     -2              /    NEEDED) FOR COLL. AND PROC.
CTT5    LAW     NPFC            /USED TO SKIP AROUND "SKAR" DATA #'S
CTT6    LAW     -2              /USED TO SWITCH BUFFERS IN COLL.
CTT7    0                       /TELLS # OF UNWANTED STOPS IN COLL.
CTT8    LAW     -DATSTR         /COUNTS # OF DATA #'S PER BUFFER
CTT9    LAW     -1              /ALLOWS PROC. TO READ DATA
CTT10   0                       /TELLS PROC TO RESTART A FILE 1
CTT11   0                       /TELLS PROC.  END OF RUN
ALF     51004                   /USED TO PUT A LINE FEED IN THE
        20100                   /    MIDDLE OF A FORTRAN OUTPUT LINE
TST1    0                       /SAVE ID TO TEST FOR COLL. STORAGE
SAVAC   0                       /SAVES AC DURING PROC. OPERATION
SAVEAC  0                       /SAVES AC FOR API LEVEL 6
TC10    LAW     -2              /USED TO UNPACK 2 WORDS
CN1     0                       /STORES # OF FRAMES FOR ONE FILE
CN2     0                       /STORES # OF FRAMES FOR ONE DISK
CNT     0                       /ODD FOR O-FRAMES, EVEN FOR X-FRAMES
CNT2    0                       /SET TO -1 FOR O-FRAMES
DUM1    0                       /GETS & STORES 1 LINEARIZED DATA WORD
DUM2    0                       /STORES THE MAX. NOISE IN A FRAME
DUM3    0                       /STORES THE SUM OF THE NOISE PER FRAME
DUM4    0                       /STORES THE MAX. OF 45 NOISE SAMPLES
DUM5    0                       /STORES THE SUM OF 45 NOISE SAMPLES
MAX4    777                     /STORES THE SMALLEST MAX. OF "DUM4"
TBUF    .DSA    BUF1            /STORES NAME OF BUF. TO BE OUTPUTED
IDCOU   0                       /STORES THE ID NUMBER
TRANF   0                       /DATA IS TRANSFERED WHEN NON-ZERO
COUNTP  LAW     -NDPC           /COUNTS # OF DATA PER FRAME TO PACK
BPOINT  .DSA    BUF             /POINTER FOR "BUF"
COUNT   LAW     -NDPC           /COUNTS # OF POINTS TO UNPACK
ER2     0                       /STORES TIMING ERROR FOR A-D CONVERTER
INFLAG  0                       /TIMING ERROR FLAG
INSUB   0                       /STORES R.-T. SUB. & API LEVEL FOR A/D
TIME    0                       /STORES TIME (RESET EVERY .1 MIN.)
TIMR    0                       /STORES TIME USED IN DATA HEADING
SVAC    0                       /SAVES AC DURING TIMER ROUTINE (TOD)
MIN     0                       /STORES THE MIN. IN DURING EACH HOUR
MCNTR   0                       /STORES # OF MIN. OF RUN
LETCHG  .DSA    NBLK            /ADDRESS OF SUB. "READM" BUFFER
BUF11   .DSA    BUF1            /ADDRESS OF BUF1
BUF21   .DSA    BUF2            /ADDRESS OF BUF2
BUF31   .DSA    BUF3            /ADDRESS OF BUF3
SEVN    370000                  /13[illegible] DECIMAL
        .DEC
MMIN    599                     /MAX # OF .1 MIN.'S IN 1 HOUR
CHR     23599                   /MAX. AMOUNT OF TIME IN ONE DAY
[illegible]HR   1000-599        /USED TO INCREMENT TIME BY 1 HOUR
        .OCT
DBP     .DSA    DBO             /POINTER FOR THE DB'S
DBO     0                       /DB'S STORAGE BLOCK
        30540                   /ASCII DEFAULT SETTINGS
        31152                   /    ARE 0,10,25 (DEC)
        LAW     -1              /TELLS END OF DB SETTINGS
        0
ST1     .BLOCK  1000            /LIN. TABLE
POINT   .DSA    SEVN            /POINTER FOR COLLECTION BUFFERS
BUF1    .BLOCK  DTBLK-323               /FIRST COLLECTION BUFFER
STT     .SIXBT  "TABLEADAT"     /NAME OF FILE TO BE READ IN
MSG7    2000
        0
LAST    .ASCII  "SET CONSOLE SWITCHES, TURN ON PULSER AND ENCODE PU"
        .ASCII  "LSE"<15>
MSG8    2000
        0
        .ASCII  "DISCONNECT WIRE FROM PULSED OSCILLATOR"<15>
MSG9    2000
        0
        .ASCII  "SET ATTENUATOR TO 00DB"<15>
MSG10   2000
        0
        .ASCII  "TYPE # OF HOURS FOR THE RUN AND C.R."<15>
MSGDB   2000
        0
```

```
        .ASCII  "DB SETTING"<15>
ERRCAL  2000
        0
        .ASCII  "ERROR IN CALIBRATION TABLE"<15>
DMV1    JMP     BLK2                    /USED TO DELETE COLL. READ
DMV2    204                             /LOC. TO DETERMINE WHICH TTY CONFIRMS ↑C
DMV3    177                             /LOC. USED TO ALLOW SHARE
DMV4    116                             /ADDR. OF LOC. OF FOREGND .DAT SLOT 0
DMV5    117                             /ADDR. OF LOC. OF BACKGND .DAT SLOT 0
DMV6    113                             /ADDR. OF THE .IOIN TABLE
SRPO    0                               /USED TO SURPRESS CALIBRATION
FDAT1   0                               /THESE LOCATIONS ARE USED
FDAT2   0                               /    TO STORE THE
FDAT3   0                               /    FOREGROUND .DAT SLOT
FDAT5   0                               /    ADDRESSES FOR 1,2,3,4,5
FDAT10  0                               /    AND 10
MINEQC  .DSA    MINEQ                   /USED TO DETERMINE LENGTH OF RUN
        .DEC
MINEQ   6000                            /CONVERTS INPUT NUMBERS TO
        600                             /    THE EQUIVALENT
        60                              /    BCD # OF MINUTES
        .OCT
CNTR1   0                               /USED TO DETERMINE MULTIPLES OF 10
DBP1    0                               /TEMP. STORAGE FOR DB SETTING
DBP2    0                               /TEMP. STORAGE FOR DB IN ASCII CODE
DBCNT1  0                               /COUNTER FOR # OF DB TO BE USED
DBCNT2  LAW     -2                      /ALLOWS NO MORE THAN 2 DIGIT DB'S
DBCNT3  LAW     -4                      /ALLOWS ONLY 4 DB SETTINGS
CODE1   777723                          /-55 TO CECK FOR CARRIAGE RETURN
CODE2   .DSA    MINEQ+2                 /FOR LESS THAN 10 HOUR RUN
CODE3   10060                           /ASCII SPACE AND A ZERO
CODE4   15                              /CHECK FOR CARRIAGE RETURN
CODE5   LAW     -2                      /CHECKS FOR MULTI-DB SETTINGS
CODE6   20140                           /ASCII DEFAULT DB FOR 0 DB SETTING
CODE7   7700                            /USED TO FIND THE DEVICE # IN .IOIN
CODE8   777766                          /-12 TO FIND BCD # MULTIPLE OF 10
CODE9   72                              /SETS THE DB SETTING
CODE10  60                              /    TO ASCII CODE
CODE11  11                              /INITIALIZES .IOIN TABLE POINTER
CODE12  12                              /USED TO DETERMINE VALUES OF ASCII #'S
CODE13  1200                            /LOOKS FOR DEVICE # 5 (DISK)
CODE14  1000                            /LOOKS FOR THE DEVICE # 4 (DECTAPE)
CODE15  160000                          /USED TO FIND THE UNIT # FOR A DEVICE
CODE16  16                              /CHECKS FOR DECIMAL POINT
CODE17  LAW     -32                     /USED TO CHECK FOR NONNUMBER ASCII CHAR.
CODE18  -MXFPD                          /SETS MAXIMUM AMOUNT COLL. ON A DISK
CODE19  .DSA    DECN                    /INIT. MULTIPIERS FOR BCD #'S
CODE20  .DSA    MINEQ                   /INIT. MULT. TO CONVERT TIME TO MIN

CODE21  NOFDK                           /SETS UP # OF DISKS TO BE USED
CODE22  20000                           /CHECKS FOR UNIT 1
CODE23  60000                           /CHECKS FOR UNIT 3
CODE24  -310000                         /CHECK FOR ASCII "1" IN DATE
CODE25  -14000                          /USED TO CHECK FOR ASCII "5" TO "7"
CODE26  3760                            /USED TO GET SECOND CHAR. IN ASCII WORD
CODE27  1400                            /CHECKS FOR AN ASCII ZERO
CODE28  360000                          /USED TO CHECK FOR AN ASCII "A"
CODE29  -10000                          /USED TO INCREMENT 2 ASCII LETTERS
CODE30  -20000                          /USED TO INCREMENT 4 ASCII LETTERS
CODE31  -4000                           /USED TO INCREMENT 1 ASCII LETTER
CODE32  20200                           /CHECKS FOR AN ASCII "A"
CODE33  7                               /USED TO MASK ALL HIGHER BITS
CODE34  3                               /CHECKS FOR FIRST HALF ASCII "Y"
CODE35  -27340                          /DEFAULT VALUE FOR TIME OF DAY (20 HR.)
CODE36  -100000                         /CHECKS FOR UNITS LESS THAN 4
CODE37  101204                          /DUMMY CODE FOR .IOIN TABLE
CODE38  402574                          /SECOND WORD FOR .IOIN TABLE
CODE39  1600                            /LOOKS FOR DEVICE #7 (P. PUNCH)
CODE40  200300                          /USED TO GET CONSOLE SWITCH 1
/''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        .EJECT
/$$$$$$$$$$ $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
/        BEGINNING OF THE PARTIAL REFLECTION PROGRAMS.
/    THIS PART INITIALIZES THE .DAT SLOTS, SETS UP NEW
/DIRECTORIES, ALLOWS SHARE MODE, AND SETS AND INITIATES THE
/CLOCK TO GIVE THE TIME OF DAY.
/
INTIM   .INIT   TTO,OUT,INTIM                   /TTY OUT
        .INIT   TTI,IN,INTIM                    /TTYIN
        LAC*    DMV4                    /GET ADDR. OF FOREGND .DAT SLOTS
        IAC
        DAC     FDAT1                   /     ONE
```

```
        IAC
        DAC     FDAT2               /    TWO
        IAC
        DAC     FDAT3               /    THREE
        IAC
        IAC
        DAC     FDAT5               /    FIVE
        TAD     CODE34
        DAC     FDAT10              /    EIGHT
        LAC*    FDAT2               /SET .DAT SLOT 5 EQUAL
        DAC*    FDAT5               /    TO .DAT SLOT 2
        LAC*    FDAT3               /SET .DAT SLOT 1 EQUAL
        DAC*    FDAT1               /    TO .DAT SLOT 3
        .INIT   OUTPT,OUT,INTIM          /PREPARE FIRST STORAGE DEVICE
        .CLEAR  OUTPT
        LAW     -2                  /ARE TWO STORAGE
        TAD     CODE21              /    DEVICES
        SPA                         /    REQUIRED ?
        JMP     NDK11               /NO, OMIT FOLOWING CODE
        LAC     MDUM1               /YES, CLEAR JUMP AROUND INSTRUCTIONS
        DAC     NDK2                /    TO USE A SECOND DISK

        DAC     NDK3                /    FOR STORAGE
        .INIT   DATIN2,OUT,INTIM         /PREPARE SECOND DEVICE FOR USE
        .CLEAR  DATIN2              /REMOVE ALL FILES FROM IT
        JMP     .+3
NDK11   DZM*    FDAT1               /CLEAR .DAT SLOTS FOR THE
        DZM*    FDAT3               /    SECOND STORAGE DEVICE
        LAS                         /GET THE VALUE OF CONSOLE
        AND     CODE40              /    SWITCH #1
        DAC     SRPO                /SET ADDR. TO THIS VALUE
        LAW     -1
        DAC*    DMV3                /ALLOW SHARE
        DZM*    DMV2                /SET TT0 TO ACKNOWLEDGE ↑C
        JMP     RDERR               /JUMP AROUND SECOND BUFFER
/............................................................
BUF2    .BLOCK  DTBLK-350           /SECOND COLLECTION BUFFER
/............................................................
RDERR   LAC     CODE19              /INITIALIZE BCD
        DAC     HRC                 /    POINTER
        ISZ     CTM1                /HAS FOUR NUMBERS BEEN
        SKP                         /    READ
        JMP     OTLP                /YES, EXIT FROM ROUTINE
        .WRITE  TTO,ASC,MSGT,0      /NO, ASK FOR TIME
        .WAIT   TTO
        JMS     READM               /READ IN TIME
DT      0                           /CONTAINS THE ADDR. OF CHAR. READ IN
        LAW     -5                  /INITIALIZE COUNTER TO EXIT THE
        DAC     CTM1                /    ROUTINE AFTER FIVE #'S
        LAW     -3                  /INITIALIZE COUNTER TO GET
        DAC     CTM2                /    THE MINUTES
        DZM     MIN                 /INITIALIZE LOC. THAT SAVE
        DZM     HR                  /    THE TIME AND MINUTES
NXT1    JMS     CHKN                /GET NEXT NUMBER
        JMS*    .AD                 /MULTIPLY BY POWERS OF TEN
        LAC*    HRC                 /    TO GET BCD EQUIVALENT
        DAC     SAV2                /SAVES THE MINUTES
        TAD     HR                  /SETS UP THE NUMBERS READ IN
        DAC     HR                  /    AS THE PRESENT TIME
        ISZ     CTM2                /IS THE NUMBER PART OF THE MIN. ?
        JMP     JPAR                /NO, FIRST TWO #'S ARE THE HOURS
        LAW     -1                  /RESET COUNTER TO GET ALL THE
        DAC     CTM2                /    MIN. (REST OF THE #'S)
        LAC     SAV2                /GET MINUTES AND
        TAD     MIN                 /    SET INTO AN ADDR.
        DAC     MIN                 /    WHICH SAVES MIN.
JPAR    ISZ     HRC                 /GET NEXT MULTIPLYING #
        ISZ     CTM1                /OBTAINED 5 NUMBERS ?
        JMP     NXT1                /NO, GET NEXT NUMBER
OTLP    LAC     MMIN                /CHECK THE MINUTES
        TCA                         /IS THE MINUTES GREATER THAN
        TAD     MIN                 /    THE MAX. NUMBER OF MINUTES
        SMA                         /    IN AN HOUR ?
        JMP     RDERR               /YES, ASK FOR THE TIME AGAIN
        LAC     CHR                 /NO, CHECK THE TOTAL TIME
        TCA                         /IS THE TIME OF DAY #

        TAD     HR                  /    LARGER THAN THE BIGGEST #
        SMA                         /    ALLOW FOR THE TIME OF DAY ?
        JMP     RDERR               /YES, ASK FOR TIME AGAIN
        LAC     HR                  /NO, PUT THE TIME INTO THE
        DAC     TIMR                / ADDR. WHICH GIVE THE
        DAC     TIME                /    TIME OF DAY
        .TIMER  360,TOD,5           /SET UP THE TIMING R. T. SUB.
        JMS*    CONTL               /TRANSFER CONTROL TO CONTROL PROGRAM
        JMP     .+2
```

```
        .DSA     SRPO                 /ADDRESS FOR CALIBRATION SURPRESSION
        .IDLE
MSGT    2000
        0
        .ASCII  "TIME"<15>
CTM1    0                             /LOC. WHICH COUNTS 5 NUMBERS
CTM2    0                             /USED TO IGNORE THE HOURS
SAV2    0                             /LOC. TO SAVE THE MIN. #'S
HR      0                             /LOC. TO SAVE THE TIME
HRC     .DSA     DECN                 /POINTER FOR THE BCD MULTIPLIERS
        .DEC
DECN    10000                         /BCD MULTIPLIERS
        1000
        100
        10
        1
        .OCT
CHKN    0                             /SUB. TO SEPARATE OUT THE NUMBERS
        LAW      -72                  /PREPARE TO LOOK AT CHARACTER
        TAD*     DT                   /    READ IN
        SAD      CODE1                /IS CHAR. A CARRIAGE RETURN ?
        JMP      RDERR                /YES, CHECK FOR POSSIBLE ERROR
        ISZ      DT                   /NO,PREPARE FOR NEXT CHAR.
        SMA                           /IS ASCII CHARACTER LESS THAN 72
        JMP      .-4                  /NO, GET NEXT CHARACTER
        TAD      CODE12               /YES, IS CHARACTER
        SPA                           /    LARGER THAN 57 ?
        JMP      .-7                  /NO, GET NEXT CHARACTER
        JMP*     CHKN                 /YES, CHAR. IS A # SO EXIT
/
/$$$$$$$$$$$ $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
        .EJECT
/FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
/
RADC    0
        JMS*     .DA                  /GET VARIABLES AND PLACE ADDRESSES BELOW
        JMP      .+4                  /JUMP AROUND VARIABLES
NB1A    0                             /ADDRESS OF THE BUFFER ADDRESS
NB2C    0                             /ADDRESS OF THE WORD COUNT
NB3F    0                             /ADDRESS OF THE FLAG
        DZM*     NB3F                 /ZERO FLAG--WAIT FOR READ IN
        LAC*     NB1A                 /INSERT BUFFER ADDRESS INTO THE
        DAC      NB5                  /    A/D CALL ROUTINE
        LAC*     NB2C                 /INSERT THE WORD COUNT INTO THE
        DAC      NB4                  /    A/D CALL ROUTINE

        LAC      NB3F                 /INSERT THE FLAG ADDRESS INTO THE
        DAC      NB6                  /    A/D CALL ROUTINE
        JMS      ADREAD               /THE A/D CALL ROUTINE (TO INIT. READ):
NB4     0                             /NUMBER OF SAMPLES TO TAKE
NB5     0                             /BUF. ADDRESS IN WHICH TO STORE SAMPLES
NB6     0                             /COMPLETION AND ERROR FLAG ADDRESS
NB7     0                             /R-T SUB. FOR INT. SERV. ROUT.TO GO TO
        JMP*     RADC                 /RETURN TO FORTRAN PROGRAM
TTM     0                             /SUBR. TO WRITE LIN. TABLE ON .DAT "TBI"
        JMS*     .DA
        JMP      .+2
STA     0                             /ADDR. OF ADDR. OF LIN. TABLE
REP     LAC*     STA                  /GET ADDR. OF LIN.
        DAC      STA                  /    TABLE
        .INIT    TBI,OUT,TTM          /PREPARE STORAGE DEVICE
        .ENTER   TBI,STT              /OPEN FILE FOR TABLE
        LAC      STA                  /PUT ADDRESS OF TABLE INTO
        DAC      .+3                  /    WRITE COMMAND
        .WRITE   TBI,DUMP,0,514       /PUT TABLE ON STORAGE DEVICE
        .WAIT    TBI
        .CLOSE   TBI
        JMP*     TTM
/
/FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
        .EJECT
/+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
/
ONC4    DZM*     SUM4                 /INITIALIZE TIME OF YEAR LOC.
        LAC*     TC3                  /GET ADDRESS OF THE
        DAC      TC3                  /    DATE
        LAC*     TC3                  /GET FIRST 2 1/2 CHARACTERS
        SPA                           /ARE THEY LETTERS?
        JMP      LETMON               /YES, CHECK FOR WORDS
        TAD      CODE24               /NO, CHECK THE NUMBERS
        SPA                           /IS THE FIRST # A ONE?
        JMP      WINT                 /YES, CHECK FOR WINTER MONTHS
        TAD      CODE25               /NO, CHECK FOR SUMMER MONTHS
        SPA                           /IS THE # >4 ?
        JMP      ONC3                 /NO, EXIT
        TAD      CODE25               /YES
```

C2

```
        SMA                     /IS THE NUMBER < 8 ?
        JMP     ONC3            /NO, EXIT
        ISZ*    SUM4            /YES, SET TIME OF YEAR LOC. TO SUM.
        JMP     ONC3            /EXIT
WINT    AND     CODE26          /GET SECOND NUMBER
        SAD     CODE27          /IS THE NUMBER AN ASCII ZERO (60) ?
        JMP     ONC3            /YES, EXIT
        LAW     -1              /NO, SET TIME OF YEAR LOC.
        DAC*    SUM4            /    TO WINTER
        JMP     ONC3            /EXIT
LETMON  TAD     CODE28          /IS THE FIRST LETTER
        SPA                     /   AN "A" ?
        JMP     ONC3            /YES, EXIT
        TAD     CODE29          /NO, IS THE FIRST LETTER

        SPA                     /    A "D" ?
        JMP     WINTL           /YES, WINTER MONTH
        TAD     CODE30          /NO, IS THE FIRST LETTER
        SPA                     /    A "F" ?
        JMP     ONC3            /YES, EXIT
        TAD     CODE29          /NO, IS THE LETTER
        SPA                     /    A "J" ?
        JMP     SWDIS           /YES, LOOK AT SECOND CHARACTER
        TAD     CODE29          /NO, IS THE LETTER
        SPA                     /    A "M" ?
        JMP     EWDIS           /YES, LOOK AT THIRD LETTER
        TAD     CODE31          /NO, IS THE LETTER
        SMA                     /    A "N" ?
        JMP     ONC3            /NO, EXIT
WINTL   LAW     -1              /SET TIME OF YEAR LOC.
        DAC*    SUM4            /    TO WINTER
        JMP     ONC3            /EXIT
SWDIS   AND     CODE26          /CHECK IF SECOND LETTER
        SAD     CODE32          /    IS AN "A"
        JMP     WINTL           /YES, A WINTER MONTH
        ISZ*    SUM4            /NO, SET TIME OF YEAR LOC. TO SUMMER
        JMP     ONC3            /EXIT
EWDIS   AND     CODE33          /IS THE THIRD LETTER
        SAD     CODE34          /    A "Y" ?
        ISZ*    SUM4            /YES, SET TIME OF YEAR TO SUMMER
        JMP     ONC3            /EXIT
/
/+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
         .EJECT
/DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
/        SUBROUTINE CLEARS I/O DEVICES FROM MEMORY TO BE ABLE TO
/DISALLOW SHARING THE DATA COLLECTION DEVICES
/
FRDT    0
        LAC*    DMV6            /GET LOC. OF THE .IOIN TABLE
        DAC     DUM2            /SAVE THE LOCATION
        TAD     CODE11          /GO TO FOREGROUND DEVICES
        DAC     DUM1            /    AND SAVE THAT LOC.
        LAC*    DUM2            /GET THE NEG. OF THE # FOREGND
        TAD     CODE34          /    DEVICES AND DOUBLE THE #
        RAL                     /    (2 WORDS PER DEVICE USED)
        DAC     DUM2            /    AND USE AS THE COUNTER
REDO1   LAC*    DUM1            /GET FIRST WORD FOR THE FOREGND DEV.
        AND     CODE7           /CHECK FOR THE DEVICE #
        SAD     CODE13          /IS IT A 5 (DISK) ?
        JMP     DKS             /YES, CHECK WHICH DISK
        SAD     CODE14          /NO, IS IT A 4 (DECTAPE) ?
        JMP     DCTP            /YES, CHECK WHICH DECTAPE
        SAD     CODE39          /NO, IS IT A 7 (P. PUNCH) ?
        JMP     SAVN1           /YES, SAVE IT
DLTDS   LAC     CODE37          /DELETE ALL OTHER DEVICES
        DAC*    DUM1            /    BY INSERTING A DUMMY NAME
        TAD     CODE22          /SET UP THE NEXT DUMMY
        DAC     CODE37          /    NAME
        ISZ     DUM1            /GO TO NEXT WORD
        LAC     CODE38          /INSERT SECOND DUMMY
        DAC*    DUM1            /    NAME INTO THE TABLE
AN01    ISZ     DUM1            /GO TO NEXT DEVICE
        ISZ     DUM2            /HAS ALL THE DEVICES BEEN CHECK ?
        JMP     REDO1           /NO, CONTINUE
        JMP*    FRDT            /YES, RETURN
DKS     LAC*    DUM1            /GET FIRST WORD
        AND     CODE15          /LOOK AT UNIT NUMBER
        SAD     CODE22          /IS IT A 1 ?
        JMP     DLTDS           /YES, DELETE THE DEVICE WORDS
        SAD     CODE23          /IS IT A 3 ?
        SKP                     /YES, CHECK FOR DOUBLE DEVICE COLL.
        JMP     SAVN1           /NO, SAVE THE TWO WORDS
```

```
        LAC     NDK2                /HAS THE LOC. BEEN CHANGED
        SAD     MDUM1               /     TO USE 2 DEVICES ?
        JMP     SAVN1               /YES, SAVE
        JMP     DLTDS               /NO, DELETE LOCATIONS
DCTP    LAC*    DUM1                /GET FIRST WORD AGAIN
        AND     CODE15              /CHECK THE UNIT NUMBER
        TAD     CODE36              /IS THE UNIT # LESS
        SPA                         /     THAN 4 ?
        JMP     DLTDS               /YES, DELETE IT
SAVN1   ISZ     DUM1                /NO, GO TO NEXT WORD
        JMP     ANO1                /GO TO NEXT DEVICE
/
/DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
        .EJECT
/
CALERR  .WRITE  TTO,ASC,ERRCAL,0            /LIN. TABLE ERROR
        .WAIT   TTO
        JMS*    TBFORL              /RECALIBRATE
        JMP     CALTB
/...................................................................
BUF3    .BLOCK  DTBLK-342           /PROCESSING'S BUFFER
/%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
/
ONC3    .INIT   TBI,IN,RES          /LINEARIZATION TABLE IN
/---------------------------------------------------------------
        LAC*    DB                  /GET ADDR. OF DB SETTINGS FOR
        DAC     DB                  /     THE FORTRAN PROG.
        LAS                         /CHECK CONSOLE SWITCH
        AND     CODE40              /     #1
        SZA                         /IS IT SET ?
        JMP     ENDDB               /YES, USE DEFAULT DB SETTING
        .WRITE  TTO,ASC,MSGDB,0 /DB MESSAGE
        JMS     READM               /GET RESPONSE
DB0     0                           /LOCATION OF THE RESPONSE
CKDB    LAW     -40                 /SET ASCII CHAR. LESS THAN
        TAD*    DB0                 /     40 TO A NEG. #
        ISZ     DB0                 /PREPARE FOR NEXT CHAR.
        SPA                         /IS CHAR. > 40 ?
        JMP     ENDDB               /NO, THERE ARE NO MORE CHAR.
        TAD     CODE17              /NO, IS THE ASCII # LESS THAN

        SMA                         /     THAN 72 ?
        JMP     NUM1                /NO, USE ONLY ONE NUMBER
        TAD     CODE12              /YES, IS THE ASCII CHAR. AN
        SPA                         /     ASCII # (>57) ?
        JMP     NUM1                /NO, USE ONLY ONE NUMBER
        ISZ     DBCNT2              /YES, IS THIS THE SECOND # ?
        SKP                         /NO, CONTINUE
        JMP     NUM2                /YES, PROCESS THE TWO #'S
        DAC     DBP1                /SAVE THE FIRST #
        JMP     CKDB                /     AND GET SECOND #
NUM1    ISZ     DBCNT2              /HAS ONE # BEEN OBTAINED ?
        JMP     RESETD              /NO, RESET COUNTER
        LAC     DBP1                /YES, GET THE NUMBER AND
        DAC*    DB                  /     SET INTO FORTRAN ARRAY
        TAD     CODE3               /SET THE NUMBER UP AS A SPACE
        CLL!RAL                     /     AND A # IN ASCII FORMAT
        DAC*    DBP                 /USED TO PRINT OUT DB MESSAGE
        JMP     DBINC               /PREPARE FOR NEXT DB
RESETD  LAW     -2                  /RESET COUNTER FOR
        DAC     DBCNT2              /     2 NUMBERS
        JMP     CKDB                /TRY AGAIN
NUM2    DAC*    DB                  /SAVE SECOND # IN FORTRAN ARRAY
        TAD     CODE10              /SET UP NUMBER IN
        CLL!RAL                     /     ASCII CODE
        DAC     DBP2                /     AND STORE
        LAC     DBP1                /GET FIRST NUMBER
        JMS*    .AD                 /SET UP THE BCD
        LAC     CODE12              /     EQUIVALENT
        TAD*    DB                  /ADD TO THE SECOND #
        DAC*    DB                  /     AND STORE IN FORTRAN PROG.
        LAC     DBP1                /GET FIRST NUMBER AGAIN
        TAD     CODE10              /SET UP THE NUMBER
        SWHA                        /     IN ASCII
        CLL!RAR                     /     CODE
        TAD     DBP2                /ADD TO PRIVIOUS # TO FORM
        DAC*    DBP                 /     THE DB SETTING IN ASCII CODE
DBINC   LAW     -2                  /RESET COUNTER FOR THE
        DAC     DBCNT2              /     NEXT 2 NUMBERS
        ISZ     DB                  /NEXT LOC. IN THE FORTRAN ARRAY
        ISZ     DBP                 /NEXT LOC. IN THE MACRO ARRAY
        ISZ     DBCNT1              /COUNTER TELLING # OF DB SETTINGS
        ISZ     DBCNT3              /HAS FOUR DB SETTINGS BEEN OBTAINED ?
        JMP     CKDB                /NO, GET NEXT SETTING
ENDDB   LAC     DBO                 /CHECK--HAS ANY NUMBERS
```

```
        SNA                       /    BEEN READ IN ?
        JMP     DEFDB             /NO, USE DEFAULT DB
        LAW     -1                /YES, ALLOW POSSIBLY ONE MORE
        DAC     DBCNT3            /    DB SETTING
        ISZ     DBCNT2            /IS ONE # STILL UNPROCESSED ?
        SKP                       /NO, CONTINUE THE EXIT
        JMP     NUM1+2            /YES, PROCESS THE LAST #
        DAC*    DBP               /SET THE LAST LOC. TO A NEG. #
        LAC     DBCNT1            /SET THE FORTRAN COUNTER
        DAC*    DBC               /    TO THE # OF DB SETTINGS
        TAD     CODE5             /IF THERE IS ONLY ONE DB
        RAL                       /    SETTING SET THE LINK
        LAC     DMV1              /SET UP JUMP AROUND COLL. READ IN
        SZL                       /IS THERE MORE THAN ONE DB SETTING ?
        DAC     UPDAT1            /NO. INSERT JUMP AROUND
        JMP     .+3               /USE INPUTED DB SETTINGS
DEFDB   LAC     CODE6             /SET UP ASCII CODE FOR
        DAC     DB0               /    ZERO DB SETTING
/-----------------------------------------------------------------
        LAC     CODE20            /NO,INITIALIZE MULTIPLIER
        DAC     MINEQC            /    TO CONVERT #'S TO MIN.
        DZM     TC1               /INITIALIZE STORAGE ADDR. FOR CONVERTING
        LAC     (DB0              /INITIALIZE DB SETTINGS TO
        DAC     DBP               /    THE BEGINING
        LAC*    DBP               /SET UP COLLECTION'S DB
        DAC     MSG9+11           /    MESSAGE
CALTB   .SEEK   TBI,STT
        .READ   TBI,DUMP,ST1,512          /READ TABLE IN
        .WAIT   TBI
        .CLOSE  TBI
        LAW     -775              /PREPARES COUNTER TO CHECK
        DAC     DUM2              /    THE VALIDITY OF THE LIN. TABLE
        LAC     (ST1+1            /SET POINTER TO THE SECOND
        DAC     DUM3              /    LOCATION OF THE TABLE
        LAC     ST1               /GET THE FIRST NUMBER AND
CALLP   TCA                       /    COMPLIMENT
        TAD*    DUM3              /IS THE PREVIOUS NUMBER
        SPA                       /    LARGER?
        JMP     CALERR            /YES, ERRONEOUS TABLE
        LAC*    DUM3              /GET PRENSENT LIN. NUMBER
        ISZ     DUM3              /GO TO NEXT NUMBER
        ISZ     DUM2              /IS IT THE END OF THE TABLE ?
        JMP     CALLP             /NO, CHECK NEXT NUMBER
        DZM*    FDAT10            /YES, DELETE .DAT SLOT 10
        LAS                       /GET THE # 1 DATA SWICH FROM
        AND     CODE40            /    CONSOLE
        SZA                       /IS IT SET ?
        JMP     DLWT              /YES, SURPRESS PRINT OUT
        .WRITE  TTO,ASC,MSG7,0
        .WRITE  TTO,ASC,MSG8,0
        .WRITE  TTO,ASC,MSG9,0
DLWT    .WRITE  TTO,ASC,MSG10,0 /NUMBER OF HOURS
        .WAIT   TTO
        JMS     DBSUB             /SET UP NEXT DB SETTING
        JMS     READM             /READ IN # OF HOURS
RUNTIM  0                         /CONTAINS ADDRESS OF THE #'S READ IN
        .INIT   TTO,OUT,PSTART    /RESET THE ↑P RESTART
        .INIT   TTI,IN,PSTART     /    ADDRESS
        LAC     CODE35            /DEFAULT VALUE  FOR THE LENGTH
        DAC     TC3               /    OF RUN (20 HOURS)
        DZM     DSTOR             /CHECKS # OF #'S READ IN
        LAW     -3                /PREPARE TO USE NO MORE
        DAC     TC2               /    THAN THREE NUMBERS
GETHR   LAW     -40               /CHECK IF CHARACTER READ IN
        TAD*    RUNTIM            /    IS AN ASCII # OR PERIOD
        DZM*    RUNTIM            /RESET LOC. IN CHAR. BLOCK
        ISZ     RUNTIM            /GO TO NEXT CHARACTER
        SPA                       /IS ASCII CHARACTER < 40 ?
        JMP     FNUM              /YES, EXIT
        SAD     CODE16            /NO, IS IT A DECIMAL POINT ?
        JMP     DECPT             /YES, GO TO DECIMAL POINT ROUTINE
        TAD     CODE17            /NO, CHECK FOR AN ASCII NUMBER
        SMA                       /IS THE CHAR. < 72 ?
        JMP     GETHR             /NO, GET NEXT CHAR.
        TAD     CODE12            /YES, IS CHARACTER GREATER
        SPA                       /    THAN 57 OCTAL ?
        JMP     GETHR             /NO, GET NEXT CHARACTER
        JMS*    .AD               /YES, GET THE BINARY CODED DECIMAL
        LAC*    MINEQC            /    EQUIVALENT OF THE NUMBER
        TAD     TC1               /ADD TO THE PRECEEDING NUMBERS
        DAC     TC1               /    AND SAVE
        ISZ     MINEQC            /SET UP NEXT MULTIPLIER
        ISZ     DSTOR             /INCREMENT NUMBER COUNTER
        ISZ     TC2               /HAS ENOUGH #'S BEEN OBTAINED ?
        JMP     GETHR             /NO, GET NEXT NUMBER
```

```
        JMP     CNUM            /YES, EXIT
DECPT   LAW     -1              /SET COUNTER TO GET ONLY
        DAC     TC2             /   ONE MORE NUMBER
        LAW     -2              /CHECK THE NUMBER
        TAD     DSTOR           /    COUNTER
        SMA                     /HAS TWO NUMBERS BEEN OBTAINED ?
        JMP     GETHR           /YES, GET THE LAST NUMBER
        LAC     TC1             /NO, IF THERE WAS A NUMBER, IT IS
        CLL                     /    OFF BY A FACTOR OF TEN TOO HIGH
        IDIV                    /THEREFORE REDUCE THE NUMBER BY A
        12                      /    FACTOR OF TEN (ZERO IS UNEFFECTED)
        LACQ                    /GET THE QUIOTIENT
        DAC     TC1             /REPLACE WITH CORRECTED #
        LAC     CODE2           /SET THE BCD POINTER TO
        DAC     MINEQC          /    THE LAST MULTIPLIER
        ISZ     DSTOR           /INCREMENT NUMBER COUNTER
        JMP     GETHR           /GET NEXT NUMBER
FNUM    LAW     -2              /CHECK THE NUMBER
        TAD     DSTOR           /    COUNTER
        SMA                     /HAS 2 NUMBERS BEEN READ IN ?
        JMP     CNUM            /YES, IGNORE THE FOLLOWING CODE
        IAC                     /HAS EVEN ONE NUMBER BEEN
        SZA                     /    READ IN ?
        JMP     NONUM           /NO, USE THE DEFAULT VALUE (13.65 HR)
        LAC     TC1             /YES, THERE IS ONE NUMBER BUT
        CLL                     /    IT IS THE WRONG BCD
        IDIV                    /    EQUIVALENT SO REDUCE IT BY
        12                      /    A FACTOR OF TEN
        LACQ                    /GET THE INTEGER ANSWER
        DAC     TC1             /SAVE CORRECTED NUMBER
CNUM    LAC     TC1             /GET LENGTH OF RUN NUMBER
        TCA                     /    COMPLIMENT IT
        DAC     TC3             /    AND SAVE THE NEGATIVE
NONUM   DZM     TC1             /INITIALIZE LOC. TO DETERMINE
        DZM     TC2             /    THE END OF THE RUN
        DZM     MCNTR           /INITIALIZE BINARY TIME OF DAY
        LAC     CODE18          /SET UP MAX. STORAGE FOR
        DAC     DSTOR           /    STORAGE DEVICE
        JMS     FRDT            /FREE BACKGROUND DEVICES
        .TIMER  0,BEGIN,6       /SET UP MAIN REAL TIME SUB.
        JMS*    PROC            /START PROCESSING PROG.
        JMP     .+4             /# OF PARAMETERS +1
        .DSA    SUM4            /THE SEASON OF THE YEAR
        .DSA    ALF             /LINE FEED
        .DSA    TIMR            /TIME AT THE END OF EACH FILE
        .IDLE
/
/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
END     2000
        0
        .ASCII  <14>"***END OF PROCESSING***"<15>
STPM    2000
        0
        .ASCII  <7><7><15>
        .EJECT
/RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
/       THIS SUBROUTINE IS USED TO READ IN CHARACTERS FROM THE
/TELETYPE ONE AT A TIME UNTIL A CARRIAGE RETURN IS FOUND.  THE
/CHARACTERS ARE STORED IN "NBLK" (A BLOCK OF LENGTH 30).  THE
/ADDRESS OF "NBLK" IS PUT INTO THE LOCATION AFTER THE "JMS   READM"
/INSTRUCTION USED TO CALL THIS ROUTINE.  THE PROGRAM CONTROL IS
/THEN RETURNED TO TWO LOCATIONS AFTER THE CALLING INSTRUCTION.
/
READM   0
        LAC     NUMB            /SET THE ADDR. OF THE BLOCK CONTAINING
        DAC*    READM           /    THE READ IN CHAR. INTO THE CALLING
        ISZ     READM           /    LOC.+1 AND RETURN TO LOC.+2
CONR    .READ   TTI,IA,MTTI,3   /READ IN ONE CHARACTER
        .WAIT   TTI
        LAC     MTTI+2          /GET THE CHARACTER
        DAC*    NUMB            /STORE IN THE CHAR. BLOCK
        SAD     (177            /IS THE CHAR. A RUBOUT ?
        JMP     ROU             /YES, DELETE PRIVIOUS CHAR.
        ISZ     NUMB            /NO, PREPARE FOR NEXT CHAR.
        SAD     (25             /IS THE CHAR. A ↑U ?
        JMP     DLT             /YES, DELETE THE LINE
        SAD     (15             /NO, IS THE CHAR. A CARRIAGE RETURN ?
        SKP                     /YES, EXIT FROM THE READ LOOP
        JMP     CONR            /NO, GET NEXT CHAR.
        LAC     (NBLK           /INITIALIZE CHARACTER BLOCK
        DAC     NUMB            /    POINTER
        LAC     (12             /WRITE OUT ON TTY
        DAC     RBOT+2          /    A LINE FEED
```

```
        JMS     WRT                     /ONE LETTER WRITE ROUTINE
        JMP*    READN                   /RETURN TO CALLING ADDR.+2
DLT     LAC     (NBLK                   /RESET POINTER TO THE BEGINING
        DAC     NUMB                    /     OF THE BLOCK
        LAC     (100                    /SET UP TO WRITE
        JMP     .+11                    /     AN @ CHAR.
ROU     LAC     (15                     /SET A CARRIAGE RETURN INTO THE
        DAC*    NUMB                    /     PRESENT LOC. OF THE CHAR. BLOCK
        LAW     -1                      /GO BACK TO WRITE OVER
        TAD     NUMB                    /     THE PREVIOUS CHAR.
        SAD     (NUMB                   /IS THE POINTER AT THE BEGINING ?
        JMP     CONR                    /YES, DO NOT PRINT OUT ANYTHING
        DAC     NUMB                    /NO, RESET POINTER BACK 1 LOC.
        LAC     (134                    /PREPARE TO WRITE A "RUBOUT" CHAR.
        DAC     RBOT+2                  /SET IN OUTPUT ADDR.
        JMS     WRT                     /WRITE OUT THE ONE CHAR.
        JMP     CONR                    /READ NEXT CHAR.
WRT     0
        .WRITE  TTO,IA,RBOT,3           /WRITE ONE IMAGE ALPHANUM. CHAR.
        .WAIT   TTO
        JMP*    WRT                     /RETURN
RBOT    2003                            /IMAGE A. OUT PUT
        0                               /     MESSAGE
        0                               /LOC. TO STORE CHAR. TO OUTPUT
MTTI    .BLOCK  3                       /BLOCK CHAR. IS READ INTO
NUMB    .DSA    NBLK                    /CHARACTER BLOCK POINTER
NBLK    .BLOCK  30                      /BLOCK TO STORE ALL THE READ IN CHAR.
/
/RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
/TTTTTTTTT TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
/       THE R.-T. SUBROUTINE THAT KEEPS TRACK OF THE TIME OF DAY.
/THE SUBROUT INE IS INITIATED BY A ".TIMER" WHILE STORING THE PRESENT
/TIME IN LOCATION "TIME" AND THE MINUTES * TEN IN "MIN".  THE ROUTINE
/RESTARTS IT SELF STORING THE UPDATED TIME IN "TIME" AND THE UPDATED
/MINUTES TIMES TEN IN "MIN" USING BINARY CODED DECIMAL.  THE BINARY
/REPRESENTATION OF THE NUMBER OF TENTHS OF MINUTES THAT HAS PAST IS
/GIVEN IN LOCATION "MCNTR".
/
TOD     0
        DAC     SVAC                    /SAVE AC FROM LOWER APL LEVELS
        .TIMER  360,TOD,5               /RESTART CLOCK FOR ANOTHER 6 SECONDS
        ISZ     MCNTR                   /INCREMENT BINARY COUNTER
        LAC     MIN                     /CHECK THE MINUTE ADDR.
        SAD     MMIN                    /IS IT SET TO 59.9 MINUTES ?
        JMP     CMIN                    /YES, SET UP THE NEW HOUR
        ISZ     MIN                     /NO, INCREMENT THE MINUTE COUNTER
        ISZ     TIME                    /     AND TIME OF DAY COUNTER
RESTOR  LAC     SVAC                    /RESTOR AC FOR LOWER LEVEL PROGHAM
        .RLXIT  TOD                     /EXIT FROM PHOG. AND API LEVEL
CMIN    DZM     MIN                     /RESET MINUTES TO ZERO
        LAC     TIME                    /CHECK TIME OF DAY
        SAD     CHR                     /IS THE TIME AT THE END OF THE DAY ?
        JMP     CTIME                   /YES, START A NEW DAY
        TAD     NXHR                    /NO, CHANGE TO NEXT HOUR
        DAC     TIME                    /     AND SAVE THE TIME
        JMP     RESTOR                  /EXIT
CTIME   DZM     TIME                    /RESTART TIME TO A NEW DAY
        JMP     RESTOR                  /     AND EXIT
/TTTTTTTTT TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
        .EJECT
PSTART  DZM     BEGIN                   /RESET R.-T. SUB. BEGIN TO BE REENTERED
        LAC     MDUMI                   /PREPARE TO REINITIALIZE THE STORAGE
        DAC     START                   /     DEVICE AND BEGIN A NEW FILE
        LAC     IDCOU                   /HAS THE COLLECTION FINISHED COLLECTING
        SPA                             /     THE PRESENT FILE ?
        JMP     .+3                     /YES, LEAVE AS SET UP
        LAC     (JMP    RESTAR          /NO, PREPARE TO CLOSE THE
        DAC     START                   /     OLD FILE
        .TIMER  0,BEGIN,6               /RESTART THE COLLECTION
        JMP     DUM7+1                  /RESTART THE PROCESSING PROGRAMS
/MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
/   MAIN PROGRAM
/
BEGIN   0                               /ENTERANCE TO THE MAIN REAL TIME SUB.
        DAC     SAVEAC
START   NOP                             /USED TO CONTROL PROGRAM FLOW
C1      .INIT   OUTPT,OUT,RESTAR              /DT OUT
        JMP     INIT                    /GO TO INITIALIZING ROUTINE
RDO     JMS     ADREAD                  /PREPARE TO READ FROM A-D CONVERTER
        TNSAM                           /# OF DATA NUMBERS TO READ
        BUF                             /ADDR. TO STORE DATA #'S
        ER2                             /COMPLETION AND ERROR TIMING FLAG
        500000+RSUB                     /PRIORITY LEVEL + R-T SUBR. TO EXEC.
```

```
RET3    LAW     -1              /LOAD -1 INTO MEMORY TO KEEP TRACK OF
        DAC     CNT2            /     THE SAMPLE THAT HAS BEEN COLL.
        LAC     SAVEAC          /RESTORE ACCUMULATOR
END1    .RLXIT  BEGIN           /RELINQUISH CONTROL TO LOWER PRIORITY
ONCE    NOP
        .TIMER  9,CHECK,5       /SET CLOCK TO WAIT 9/60 SECONDS
PKO     JMS     ADREAD          /A-D CONV. READ FOR X-SAMPLES
        TNSAM                   /THE VARIABLES USED
        BUF                     /    ARE THE SAME ONES USED FOR
        ER2                     /    THE O-SAMPLE AND ARE
        500000+RSUB             /    EXPLAINED ABOVE
        LAC     SAVEAC
END2    .RLXIT  BEGIN           /RELINQUISH CONTROL
RET2    LAC     TRANF           /GET DECTAPE TRANSFER FLAG
        SZA                     /    AND TEST IT
        JMS     DTRANS          /FLAG SET - TRANSFER DATA
A       CLA!CLL                 /CLEAR AC TO READ CONSOLE SWITCHES
        LAS                     /GET # FROM CONSOLE'S DATA SWITCHES
        RAL                     /PUT BIT 00 OF AC IN L
        SNL                     /IS THE LINK A ZERO ?
        JMP     RDO             /YES, L=0 COLLECT DATA
        .TIMER  120,WT1,6       /NO, STOP COLL. AND GIVE TIME TO A
        .RLXIT  BEGIN           /    LOWER LEVEL BEFORE RECHEKING SWITCH
WT1     0
        DAC     SAVEAC
        LAC     MDUM1           /PUT CLOCK BACK INTO
        DAC     ONCE            /    OPERATION
        DZM     CNT             /TELLS PROC. COLL. HAS STOPPED
        LAC     (JMP    A       /SET UP TO RECHECK CONSOLE SWITCHES
        DAC     START           / AND PUT INTO START
        LAC     SAVEAC
        .TIMER  0,BEGIN,6       /RETURN TO R.-T. SUB. BEGIN
        .RLXIT  WT1
/
/  END OF MAIN PROGRAM
/MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
BUF     .BLOCK  DATBLK          /BUFFER TO STORE AD SAMPLES
NAME    .SIXBT  "DATAF1DAT"     /NAME OF FILE TO STORE DATA
/&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
/  INITIALIZING ROUTINE
/
INIT    LAC     (BUF1
        DAC     POINT           /POINTER IN DECTAPE BUFFER
        DAC     TBUF            /NAME OF DECTAPE BUFFER IN USE
        LAC     IDCOU           /SAVE THE ID # (THE #
        DAC     CN1             /    OF FRAMES PER FILE)
        DZM     IDCOU           /ID NUMBER
        DZM     TRANF           /DECTAPE TRANSFER FLAG
        DZM     CNT             /COUNTER FOR CLOCK
        LAC     MDUM1           /PUT CLOCK INTO OPERATION
        DAC     ONCE
        LAW     -2              /SET BUF1 AS THE FIRST
        DAC     CTT6            /    BUFFER TO BE USED
        LAC     (777            /INIT. NOISE MAX.
        DAC     MAX4            /    LOCATION
        LAW     MNCC            /INITIALIZE COUNTER FOR MAXIMUM
        DAC     CTT2            /    ALLOWABLE NOISE
        LAC     MCNTR           /GET THE LENGTH OF TIME REQUIRED
        TAD     TC1             /    COLLECT THE PRIVIOUS FILE
        DAC     DBC             /SAVE IT
        TAD     TC2             /ADD TO ALL OTHER PRIVIOUS TIMES TO
        DAC     TC2             /    COLLECT THE OTHER FILES, SAVE AND
        TAD     TC3             /    COMPARE TO THE MAXIMUM TIME
        SMA                     /ARE THE FILE TIMES LARGER ?
        JMP     EXTIP           /YES, STOP COLLECTION
/
/  DECTAPE FILE ROUTINE
/  STORE DATA IN FILE ACCORDING TO RESPONSE
/
UP      .FSTAT  OUTPT,NAME      /CHECK FOR COLLECTED FILE
UP1     SKP                     /(REPLACED BY "SZA")IS FILE PRESENT?
        JMP     UPDATE          /YES, ACKNOWLEDGE THE PRESENCE
WRITE   LAC     MCNTR           /GET THE BEGINNING TIME FROM
        TCA                     /    THE BINARY CLOCK COUNTER
        DAC     TC1             /    TO DETERMINE THE COLL. TIME
        .DLETE  OUTPT,NAME      /DELETE FILE IF PRESENT
C3      .ENTER  OUTPT,NAME      /OPEN FILE
        JMS     DTRANS          /WRITE DUMMY BLOCK
        JMS     DTRANS          /TWICE
        LAC     MDUM2           /TELLS PROCESSING THAT COLLECTION
        DAC     REPL            /    HAS STARTED COLLECTING A FILE
        JMP     RDO             /RETURN
```

```
UPDATE  LAC     TIME             /SETS THE TIME TO THE END
        TAD     (5               /    OF THE COLLECTED FILE AND
        DAC     TIMR             /    AND ROUNDS OFF TO THE NEAREST MIN.
        .WRITE  TTO,ASC,MSG1,0   /FILE PRESENT
        .WAIT   TTO
UPDAT1  NOP                      /REPLACED BY "JMP  BLK2" FOR 1 DB SET.
        .WRITE  TTO,ASC,MSG2,34  /KEEP IT?
        .WAIT   TTO
BLK1    JMS     READM            /READ RESPONSE
COM     0                        /ADDRESS OF THE RESPONSE
GETCH1  LAC*    COM              /GET READ IN CHARACTER
        DZM*    COM              /    AND ZERO THE LOC.
        SAD     (116             /IS CHARACTER A "N"?
        JMP     STAGN            /YES
        TAD     (-72             /CHECK IF NUM. IS LESS
        SMA                      /    THAN 72 OCTAL
        JMP     BLK3             /NO, CONTINUE
        TAD     (12              /YES, IS NUM. GREATER THAN
        SPA                      /    57 OCTAL?
        JMP     BLK3             /NO, CONTINUE
        DAC*    COM              /YES, SAVE
        ISZ     COM              /NEXT CHARACTER
        JMP     GETCH1           /REPEAT
BLK3    LAC     LETCHG           /INITIALIZE THE BLOCK
        DAC     COM              /    CONTAINING THE INPUT CHAR.
        LAC*    COM              /GET FIRST CHAR.
        SNA                      /IS IT A ZERO ?
        JMP     CONDB            /YES, IGNORE IT
        DAC*    CHG              /PASS TO PROC THE DB CHANGE
        TAD     (-1              /OFFSET THE # BY -1
        CLL!RAR                  /DIVIDE BY TWO AND SAVE REMAINDER
        TAD     (60              /SET UP AS ASCII CHAR.
        SWHA                     /PREPARE FOR MESSAGE
        DAC*    DBP              /SAVE NUMBER
        LAC     (140             /ASCII FOR ZERO
        SZL
        LAC     (152             /ASCII FOR NUMGER 5
        TAD*    DBP              /ADD TO OTHER DB SETTING
        CLL!RAR                  /SET UP AS CHAR. 4&5 IN ASCII WORDS
        DAC*    DBP              /SAVE NEW DB SETTING
CONDB   ISZ     COM              /GET NEXT CHAR.
        LAC*    COM              /SET UP NEXT CHAR.
        SAD     (15              /IS CHAR. A CARRIAGE RETURN ?
        JMP     .+3              /YES, EXIT
        SZA                      /NO, IS IT A ZERO ?
        DAC*    CMULC            /NO, SET UP MUL. CONSTANT CHANGE
        DZM*    COM              /CLEAR CHAR.
BLK2    JMS     DBSUB            /SET UP NEXT DB MESAGE
        LAC     CN2              /GET STORAGE ALREADY USED
        TAD     CN1              /ADD STORAGE SIZE OF LAST
        DAC     CN2              /    FILE AND SAVE
        TAD     TEM1             /ADD IT AGAIN AND CHECK--
        TAD     DSTOR            /    WILL ANOTHER FILE OF THE SAME
        SMA                      /    LENGTH OVERFLOW THE STORAGE ALLOW?

MDUM3   JMP     RPT1             /YES, RESTART THE COUNTING
        ISZ     NAME+1           /NO, INCREMENT NAME
        ISZ     MSG1+3           /INCREMENT TELETYPE
        ISZ     MSG1+3           /    MESSAGE TWICE
        JMP     WRITE            /RETURN TO NEW FILE
STAGN   LAC     CN1              /REMOVE THE SIZE OF
        TCA                      /    THE LAST FILE
        TAD     CN2              /    FROM THE
        DAC     CN2              /    STORAGE COUNTER
        LAC     DBC              /REMOVE THE AMOUNT OF
        TCA                      /    TIME USED
        TAD     TC2              /    BY THE PRECEEDING
        DAC     TC2              /    FILE
        JMP     WRITE            /RECOLLECT FILE
/DBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDB
/       SUBROUTINE TO CHANGE THE DB MESSAGE TO BE PRINTED OUT
/
DBSUB   0
        ISZ     DBP              /GET NEXT DB
        LAC*    DBP              /    MESSAGE
        SMA                      /IS IT THE END OF THE DB MESSAGES ?
        JMP     .+4              /NO, USE THIS MESSAGE
        LAC     (DB0             /YES, REPEAT THE
        DAC     DBP              /    FIRST DB
        LAC*    DBP              /    SETTING GIVEN
        DAC     MSG2+11          /INSERT MESSAGE
        JMP*    DBSUB            /RETURN
```

```
/
/DBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDBDB
MSG1    MSG2-MSG1/2*1000
        0
        .ASCII  <7><14><7><7>"1"
        .ASCII  " PRESENT"<15>
MSG2    2000
        0
        .ASCII  "SET ATTENUATOR TO 00DB AND C.R."<15>
/
/&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
/$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
/  PACKING ROUTINE
/  AN ID IS ASSIGNED AND 26 SAMPLES ARE PACKED TWO PER WORD
/  A TOTAL OF 14 WORDS ARE PUT INTO A DECTAPE BUFFER FOR EACH CALL
/  HIGHEST ORDER BIT IS LOST AND NEG. NUMBERS ARE SET TO ZERO
/  ALTERNATES BETWEEN TWO DECTAPE BUFFERS-
/  WHEN ONE IS FULL THE OTHER IS USED FOR STORAGE
/  THE BUFFERS ARE LOCATED IN DLOGF AND WRITE OVER PART OF IT
/
RSUB    0
        DAC     SAV                     /SAVE AC
        ISZ     CNT                     /INCREMENT COUNTER
MDUM1   NOP                             /GUARDS AGAINST A -1 IN "CNT"
        DZM     DUM2                    /INITIALIZE THE VARIABLES USED
        DZM     DUM3                    /     TO DETERMINE THE NOISE
        LAW     NPFC                    /USE FIRST 5 NOISE SAMPLES TO SET

        DAC     CTT1                    /     MAX. ALLOWED NOISE FOR PROC.
        IAC                             /SKIP NOISE AFTER THE 5TH
        DAC     CTT5                    /     NOISE SAMPLE
        LAW     -1001                   /IS THERE A DISK TIMING
        SAD     ER2                     /     ERROR?
        JMP     ERS1                    /YES, PRINT MESSAGE
NOISE   ISZ     IDCOU                   /NO, INCREMENT ID NUMBER
        LAC     IDCOU                   /ASSIGN ID NUMBER
        DAC*    POINT                   /      TO BE PLACED ON STORAGE DEVICE
        ISZ     POINT                   /INCREMENT POINT TO NEXT LOCATION
DATA    LAW     -NDPC                   /COUNTER SO THAT 13 WORDS ARE PACKED
        JMS     PAC                     /GO TO PACKING ROUTINE
        LAC     DUM3                    /GET SUM OF THE NEW 4 NOISE SAMPLES
        TAD     DUM5                    /   AND ADD TO THE 72 NOISE SAMPLE GROUP
        DAC     DUM5                    /USE NEW SUM
        LAC     DUM2                    /GET NEW MAX. FOR THE 4 NOISE SAMPLES
        TCA                             /COMPARE NEW MAX.
        TAD     DUM4                    /     WITH OLD MAX.
        SMA                             /IS THE NEW MAX. LARGER?
        JMP     .+3                     /NO, KEEP THE OLD MAX.
        LAC     DUM2                    /YES, REPLACE THE OLD MAX.
        DAC     DUM4                    /      WITH THE NEW MAX.
        ISZ     CTT2                    /HAS 72 NOISE SAMPLES BEEN COLLECTED?
        JMP     SKP4                    /NO, PREPARE TO COLLECT MORE SAMPLES
COMP    LAW     MNCC                    /RESET COUNTER FOR
        DAC     CTT2                    /     NOISE DETERMINATION
        LAC     DUM4                    /YES, GET MAX. OF NEW 72 NOISE SAMP.
        TCA                             /     AND COMPARE IT WITH THE MAX.
        TAD     MAX4                    /     OF THE PREVIOUS SET
        SPA                             /IS THE NEW ONE LESS THAN THE OLD ONE?
        JMP     SKP3                    /NO, PREPARE FOR ANOTHER COLLECTION
        LAC     DUM4                    /YES, REPLACE THE OLD MAX. WITH
        DAC     MAX4                    /     THE NEW MAX.
        LAC     DUM5                    /REPLACE THE OLD SUM WITH
        DAC     SUM4                    /     THE NEW SUM
SKP3    DZM     DUM4                    /RESET NOISE SAMPLING
        DZM     DUM5                    /     LOCATIONS
SKP4    LAC     (JMP     ONCE           /PREPARE TO REENTER R-T SUB. "BEGIN"
        ISZ     CNT2                    /      BY INSERTING INTO LOC. START
        LAC     (JMP     RET2           /     A JUMP STATEMENT DEPENDING ON
        DAC     START                   /     CNT2 (WHICH TYPE OF SAMPLE)
        LAC     SAV
        .TIMER  0,BEGIN,6               /ENTER R-T SUB. "BEGIN"
        .RLXIT  RSUB
PAC     0
        DAC     COUNTP                  /STORE WORD NUMBER BEING PACKED
        LAC     (BUF-1
        DAC     BPOINT                  /BUF POINTER
PAKING  JMS     LIN                     /GO TO TABLE ROUTINE
        LAC     DUM1
        SWHA                            /ROTATE TO LEFT HALF
        DAC*    POINT                   /STORE
        JMS     LIN
        LAC     DUM1
```

```
        TAD*    POINT           /PACK INTO PREVIOUS ONE
        DAC*    POINT           /STORE IN BUF1
        ISZ     POINT           /MOVE POINTER UP ONE WORD
        ISZ     COUNTP          /ONE WORD HAS BEEN PACKED
        JMP     PAKIN3          /13 WORDS HAVE NOT BEEN PACKED
        LAC*    POINT           /13 WORDS HAVE BEEN PACKED
        ISZ     CTT8            /END OF BUFFER ?
        JMP*    PAC             /NO, CONTINUE COLLECTION
        LAW     -DATSTR         /YES, RESET THE BUFFER
        DAC     CTT8            /     COUNTER
        LAC     BUF21           /SET POINTER TO SECOND
        DAC     POINT           /     BUFFER--BUF2
        LAC     BUF11           /GET FIRST BUFFER ADDRESS
        ISZ     CTT6            /JUST FINISHED FILLING BUF1 ?
        JMP     .+5             /YES, PREPARE TO STORE BUF1
        DAC     POINT           /NO, SET POINTER TO BUF1
        LAW     -2              /RESET COUNTER WHICH DETERMINES
        DAC     CTT6            /     WHICH BUFFER TO TRANSFER
        LAC     BUF21           /PREPARE TO TRANSFER
        DAC     TBUF            /     BUF2
        DAC     TRANF           /SET TRANSFER FLAG
        JMP*    PAC             /CONTINUE COLLECTION
ERS1    .WRITE  TTO,ASC,MSG5,0  /TIMING ERROR
        LAC     MDUM1           /PUT CLOCK OPERATION BACK
        DAC     ONCE            /     IN PROGRAM
        ISZ     CNT             /PREPARE TO REJECT O-X PAIR
        LAC     (JMP    RDO     /REJECT NEXT X-SAMPLE
        ISZ     CNT2            /WAS THE LAST AN O-SAMPLE ?
        LAC     (JMP    ONCE    /NO, REJECT THE FORMER O-SAMPLE
        DAC     START           /YES, REJECT NEXT SAMPLE
        .TIMER  0,BEGIN,6               /RETURN TO COLLECTION
        .RLXIT  RSUB
MSG5    2000
        0
        .ASCII  "TIMING ERROR"<15>
/
/$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
/++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
/     SUBROUTINE TRANSFERS 1 BLOCK OF DATA FROM A DESIGNATED BUFFER
/  TO THE STORAGE DEVICE BEING USED.
/
DTRANS  0
        LAC     TBUF            /PASS NAME OF BUFFER
        DAC     .+3             /     TO .WRITE
C4      .WRITE  OUTPT,DUMP,0,252
C5      .WAIT   OUTPT
        DZM     TRANF           /CLEAR TRANSFER FLAG
        JMS     CKCNT           /CHECK CKCNT ROUTINE
        LAC     (NOP            /PUT CLOCK BACK INTO OPERATION
        DAC     ONCE
        JMP*    DTRANS
/
/++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
/###############################################################
/       SUBROUTINE TO CHECK WHICH DATA CONSOLE SWITCHES ARE SET.
/   TOO LARGE SETTINGS AND TOO SMALL SETTINGS ARE GUARDED
/   AGAINST.  THE DEFAULT SETTING IS 2000.
/
CKCNT   0
        LAS                     /LOAD DATA SWITCHES FROM CONSOLE
        RTL                     /PUT AC BIT 1 INTO LINK
        CLA                     /TO DISALLOW SHARING
        SZL                     /IS AC BIT 1 SET ?
        LAW     -1              /YES, DO ALLOW
        DAC*    (177            /     SHARING
        LAS                     /RELOAD DATA SWITCHES
        AND     (17777          /IGNORE TOP 5 BITS
        TAD     (-7             /ARE THE SWITCHES SET TO
        SPA                     /     LESS THAN SEVEN ?
        TAD     (2000           /YES, USE DEFAULT SETTING
        TAD     (-10000         /NO, ARE THEY SET TO GREATER
        SMA                     /     THAN 10000 (DISK OVERFLOW) ?
        TAD     (2000           /YES, USE THE DEFAULT SETTING
        TAD     (10007          /NO, RESET THE AC BACK
        DAC     TEM1#           /TEMPARILY STORE NUMBER
        CMA                     /COMPARE THE DATA SWICHES
        TAD     IDCOU           /     TO THE ID NUMBER
        SMA                     /IS THE ID LESS ?
        JMP     RESTAR          /ID GREATER THAN SWITCH SETTING
        LAC     TEM1            /WILL THE SIZE OF
        TAD     CN2             /     THIS FILE
        TAD     DSTOR           /     OVERFLOW
        SMA                     /     THE DISK ?
        JMP     RESTAR          /YES,CLOSE FILE
        JMP*    CKCNT           /ID LESS THAN SWITCH SETTING
```

```
/
/################################################################
/&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
/       ROUTINE TO CLOSE FILE AND SET UP PARAMETERS FOR
/   COLLECTION AND PROCESSING
/
RESTAR  LAC     TRANF           /COMPLETE TRANSFER IF NECESSARY
MDUM2   SZA
        JMS     DTRANS
C6      .WRITE  OUTPT,DUMP,SEVN,2       /WRITE END OF FILE ID
C2      .WAIT   OUTPT
C7      .CLOSE  OUTPT
        LAC     MDUM1           /INITIALIZE THE LOCATIONS
        DAC     START           /     "START" FOR COLLECTION
        DAC     REPL            /    AND "REPL" FOR PROCESSING
        LAC     SUM4            /GIVE THE SUMMATION OF THE LOWEST
        DAC*    SUM5            /    NOISE TO THE PROCESSING
        DZM*    CHG             /RESET DB CHANGER
        LAC     MDUM2           /IGNORE THE FIRST
        DAC     UP1             /    FILE ONLY
        JMP     START           /GO TO START
/
/&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

/<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/  CLOCK INTERRUPT ROUTINE FOR AUTOMATIC O-SAMPLE START
/
CHECK   0
        DAC     SAV             /SAVE AC
        DZM     BEGIN           /ZERO R-T SUB. TO AVOID POSS. ERROR
        LAC     CNT             /EXAMINE COUNTER
        RAR                     /LOWEST BIT OF CNT IN L
        SZL
        JMP     BKUP            /L=1-CNT ODD-HALF FRAME DURING 9/60 SEC
        JMP     NORM            /L=0-CNT EVEN-FULL FRAME DURING 9/60 SEC
BKUP    DZM     CNT             /CLEAR HALF FRAME THAT WAS TAKEN:
        RAL                     /RESTOR LINK
        LAW     -1              /RESET IDCOU BACK ONE FRAME
        TAD     IDCOU
        DAC     IDCOU
        LAW     -1              /RESET BUFFER COUNTER
        TAD     CTT8            /    BACK ONE
        DAC     CTT8            /    FRAME
        LAW     -DPR            /RESET POINT BACK HALF FRAME
        TAD     POINT
        DAC     POINT
        LAC     MDUM1           /PUT CLOCK BACK INTO OPERATION
        JMP     EXT1            /PREPARE TO EXIT
NORM    RAL                     /RESTORE THE LINK
        LAC     (JMP    PRO     /STOP CLOCK FROM OPERATION
EXT1    DAC     ONCE
        LAC     (JMP    RDO             /PREPARE TO COLLECT
        DAC     START           /    AN O-FRAME
        LAC     SAV             /RESTORE AC
        .TIMER  0,BEGIN,6       /RETURN TO POINT AT WHICH INTERRUPT
        .RLXIT  CHECK           /  OCCURED
/
/>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
/!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
/       SUBROUTINE TO DO A TABLE LOOKUP FOR DATA LINEARIZATION.
/   ALSO SAVES SUM AND MAXIMUM OF EACH SET OF 5 NOISE SAMPLES,
/   SETS NEGATIVE NUMBERS TO ZERO, AND IF NECESSARY JUMPS AROUND
/   "SKAR" NUMBER OF DATA NUMBERS BETWEEN THE NOISE AND DATA POINTS.
/
LIN     0
        ISZ     BPOINT          /DATA STARTS AT BUF
        LAC*    BPOINT          /GET INPUT DATA WORD
        AND     (1777           /MASK ANY EXTRA BITS
        TAD     (-1000          /CHECK FOR NEG. #'S
        SMA
        JMP     ER1             /NEG # FOUND
        TAD     (ST1+1000       /LOCATE # IN TABLE
        DAC     DUM1            /GET ADDRESS OF NUMBER
        LAC*    DUM1            /LOAD LINEARIZED # INTO AC
        SKP
ER1     CLA!CLL                 /SET NEG # TO ZERO
        DAC     DUM1            /STORE LINEARIZED #
        ISZ     CTT5            /IS THIS THE FIFTH NOISE SAMPLE ?
        JMP     .+5             /NO, SKIP AROUND CODE
        LAC     (SKAR           /YES, SKIP AROUND "SKAR" DATA
        TAD     BPOINT          /    OF THE FRAME BEING
        DAC     BPOINT          /    COLLECTED
        LAC     DUM1            /RESTORE THE LIN. DATA #
        ISZ     CTT1            /IS IT A NOISE SAMPLE ?
        JMP     DONOS           /YES, PROCESS NOISE SAMPLE
        LAW     -1              /RESET THE TWO COUNTERS
        DAC     CTT1            /    FOR THE DATA OF THE
```

```
        JMP*     LIN                 /    RETURN
DONOS   TAD      DUM3                /ADD TO THE OTHER 4
        DAC      DUM3                /    NOISE SAMPLES
        LAC      DUM1
        TCA                          /IS THE NOISE SAMPLE
        TAD      DUM2                /    GREATER THAN THE
        SMA                          /    OTHER 4
        JMP*     LIN                 /NO, RETURN
        LAC      DUM1                /YES, SET THIS SAMPLE
        DAC      DUM2                /    AS THE MAX.
        JMP*     LIN
/
/?????????????????????????????????????????????????????????????????
/&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
/     ROUTINE TO REINITIALIZE VARIABLES AND IF NEEDED TO
/  SWITCH STORAGE DEVICES.
/
RPT1    LAC      AF1                 /RESET FILE
        DAC      NAME+1              /    NAME
        DZM      TEM1                /ZERO LOC.
        DZM      CN2                 /RESET DEVICE STORAGE COUNTER
NDK2    JMP      NDK21               /DO NOT SWITCH STORAGE DEVICE
        LAC      (OUTPT              /CHANGE TO SECOND STORAGE
        ISZ      CTT3                /    DEVICE BY CHANGING THE
        LAC      (OUTPT2             /    .DAT SLOTS IN THE COMMANDS:
        DAC      C3                  /    .ENTER
        DAC      C2                  /    .WAIT
        DAC      C5                  /    .WAIT
        DAC      C7                  /    .CLOSE
        TAD      (1000
        DAC      WRITE               /    .DLETE
        DAC      C1                  /    .INIT
        TAD      (2000
        DAC      UP                  /    .FSTAT
        TAD      (1000
        DAC      C4                  /    .WRITE
        DAC      C6 -                /    .WRITE
NDK21   LAC      RMSG1               /RESET TELETYPE
        DAC      MSG1+3              /    MESSAGE
        LAC      MDUM4               /IGNORE  THE FIRST FILE ON
        DAC      UP1                 /   NEXT STORGE DEVICE
        LAC      CTT3                /CHECK DEVICE SWITCHING CONTROLLER
        SMA                          /IS IT STILL NEG.?
        LAW      -2                  /YES, RESET IT
        DAC      CTT3                /NO, LEAVE IT ALONE

        JMP      C1                  /NO, GO TO INIT.
AF1     .SIXBT   "AF1"               /REINITIALIZES "NAME" AND "FIL"
EXTIP   DZM      TC2                 /PREPARE TO COLLECT A NEW SET OF DATA
        LAC      TIME                /GET TIME FOR LAST FILE
        TAD      (5                  /ROUND OFF TO THE NEAREST MINUTE
        DAC      TIMR                /STORE FOR FORTRAN PROGRAM
        LAC      (JMP     RPT1       /PREPARE TO START
        DAC      START               /    COLLECTION AT FILE 1
        ISZ      NAME+1              /TELLS PROC. TO PROCESS LAST FILE
        .WRITE   TTO,ASC,EDC,0       /END OF COLLECTION
        .WAIT    TTO
        DZM      TC1                 /INIT. COLL. TIME COUNTER
        LAW      -1                  /TELLS PROCESSING THAT COLLECTION
        DAC      CTT11               /    IS FINISHED COLLECTING
        .RLXIT   BEGIN
EDC     2000
        0
        .ASCII   <11><11><11><11><11><7><7><7><7><15>
/
/&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
        .EJECT
/+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
/       BFKM15 V3A SERVICE ROUTINES FOR THE HP S610A A TO D
/  CONVERTER.  THESE ROUTINES PERMIT INPUT OF ANY SPECIFIED
/  NUMBER OF SAMPLES INTO A CORE BUFFER.  INPUT MAY BE OVER-
/  LAPPED WITH PROGRAM EXCUTION, AND CONTROL MAY BE RELINQUISHED
/  TO LOWER PRIORITY PROGRAMS WHILE DATA TRANSFER TAKES PLACE.
/    MACRO-15 CALLING SEQUENCE:
/       JMS      ADREAD
/       NUMBER OF SAMPLES REQUIRED
/       BUFFER ADDRESS
/       COMPLETION FLAG ADDRESS
/       REAL-TIME SUBROUTINE ADDRESS, PRIORITY LEVEL IN BITS 0-2
/  (EXAMPLE: 500000+RTSUBA)
/  (RETURNS HERE IMMEDIATELY)
/       IF THE 4TH WORD AFTER THE JMS IS 0, NO REAL-TIME SUBROUTINE
/  WILL BE ACTIVATED.  NOTE:  THE PRIORITY CODE FOR MAINSTREAM IS 1
/       THE COMPLETION FLAG IS CLEARED BY THE CALL TO ADREAD,
/  AND SET TO +1 FOR NORMAL COMPLETION OR -1001 IF A DATA
/  TIMING ERROR OCCURS.
/
```

```
/
ADWCR=26                                /A-D WORD COUNT
ADCAR=ADWCR+1                           /AND CURRENT ADDRESS REGISTERS
.SCOM=100                               /MONITOR'S COMMUNICATION AREA
ADWI=703724                             /A-D CONVERTER WRITE INITIALIZE
ADSO=703701                             /SKIP ON WORD COUNT OVERFLOW
ADST=703721                             /SKIP ON DATA TIMING ERROR
ADCO=703704                             /CLEAR OVERFLOW FLAG
ADCT=703744                             /CLEAR TIMING FLAG
/
/  ENTRY POINT FOR A-D INTERFACE INITIALIZATION
/
ADREAD  0

        JMP     INSET                   /REPLACED BY "LAC*     ADREAD"
        TCA
        DAC*    (ADWCR)                 /SET WORD COUNT
        ISZ     ADREAD
        LAW     -1
        TAD*    ADREAD                  /BUFFER ADDRESS -1
        DAC*    (ADCAR)                 /  TO CURRENT ADDRESS REG.
        ISZ     ADREAD
        LAC*    ADREAD                  /GET FLAG ADDRESS
        DAC     INFLAG
        DZM*    INFLAG                  /CLEAR FLAG
        ISZ     ADREAD
        LAC*    ADREAD                  /GET REAL-TIME SUBROUTINE ADDRESS
        DAC     INSUB
        ISZ     ADREAD                  /POINT TO RETURN LOCATION
        ADWI                            /INITIALIZE INTERFACE
        JMP*    ADREAD                  /RETURN
/
/   THE FOLLOWING CODE IS EXECUTED ONLY ONCE
INSET   LAC*    (.SCOM+55)              /GET ENTRY POINT ADDERSS OF .SETUP
ADSVA   DAC     .
SAV     LAC*    (.SCOM+51)              /ENTRY POINT OF REALTP
REALTP  DAC     .
        LAC     (400010                 /RAISE THE API
        ISA                             /    LEVEL
        JMS*    ADSVA                   /CALL .SETUP TO CONNECT
        ADSO                            /    ADINT TO
        ADINT                           /    THE API
        DBK                             /DEBREAK FROM API LEVEL
        LAC     (LAC*   ADREAD
        DAC     ADREAD+1                /MODIFY INSTRUCTION
        JMP     ADREAD+1                / AND JUMP TO IT
/
/
/INTERRUPT SERVICE ROUTINE.  EXECUTED IMMEDIATELY AFTER COMPLETION
/  OF DATA TRANSFER.  DETERMINES STATUS OF A-D INTERFACE, SETS
/  COMPLETION FLAG AND ACTIVATES REAL-TIME SUBROUTINE.
/  RUNS AT API LEVEL 0.
/
/
ADINT   0
        DBA                             /PAGE ADDRESSING MODE
        DAC     ADSVA                   /SAVE AC
        ADST                            /TIMING ERROR?
        SKP!CLA!IAC                     /NO,+1 TO AC
        LAW     -1001                   /YES, ERROR CODE
        DAC*    INFLAG                  /SET FLAG
        ADCO                            /CLEAR
        ADCT                            /   INTERFACE FLAGS
        LAC*    (.SCOM+102              /RAISE TO API
        ISA                             /    LEVEL 0 OR 1
        LAC     INSUB                   /REAL-TIME SUBROUTINE ADDRESS
        SNA
        JMP     ADXIT                   /BYPASS MONITOR CALLS IF ZERO

        JMS*    REALTP                  /ACTIVATE REAL-TIME SUBROUTINE
ADXIT   LAC     (404000                 /REQUEST AN API INTERRUPT
        ISA                             /    AT SOFTWARE LEVEL 4
        LAC     ADSVA                   /RESTORE AC
        DBR                             /SET TO LEAVE HARDWARE API LEVEL
        JMP*    ADINT                   /RETURN TO INTERRUPTED PROGRAM
/
/+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        .EJECT
/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
/      PROCESSING'S MACRO PROGRAMS.  THEY INTIALIZE THE SOTRAGE
/   DEVICE, WAIT FOR FILE TO BE COLLECTED, CHECK FOR UNWANTED
/   COLECTION STOPAGE, SWITCH STORAGE DEVICES IF NECESSARY, TELL
/   WHEN THE PROGRAMS HAVE REACHED THE END OF RUN, GIVES TIME TO
/   BACKGROUND, AND READS IN DATA AND UNPACKS IT.
```

```
/
CHNG    0
        JMS*    .DA                 /LOAD PARAM. ADDR. IN SUM5 AND CHG
        JMP     .+4                 /SKIP OVER PARAM. LIST
SUM5    0
CHG     0
CMULC   0                           /LOC. TO CHANGE MUL. CONSTANT
        ISZ     CTT10               /RESTART THE DEVICE ?
        SKP                         /NO
        JMP     AGN                 /YES, RESET PARAMETERS TO BEGIN. OF DEV.
DUM7    ISZ     FIL+1               /CHECK IF COLLECTION IS
        JMS     CONTR               /    FINISHED COLLECTING NEW FILE
        LAC     FIL+1               /IS THE COLLECTION
        CMA                         /    RECOLLECTING
        TAD     NAME+1              /    THE FIRST FILE
        SPA                         /    AGAIN ?
        LAW     -1                  /NO, SET THE COUNTER TO JUMP AROUND
        DAC     CTT10               /YES, SET COUNTER TO A POS. #
        LAW     -1                  /PREPARE TO READ
        DAC     CTT9                /    DATA
        LAC     BUF31               /RESET BUF3 POINTER WITH
        DAC     POINT2              /    THE ADDR. OF BUF3
        LAC     (ISZ    SWITC       /PREPARE TO READ TWO
        DAC     LCA                 /    DUMMY BLOCKS
C13     .INIT   DATIN,IN,0
RT      .FSTAT  DATIN,FIL           /CHECK FOR PRESENTS OF NEW FILE
        SNA
        JMP     ERR3                /FILE NOT PRESENT
C11     .SEEK   DATIN,FIL           /PREPARE TO READ
        JMP*    CHNG                /RETURN TO FORTRAN PROGRAM
FIL     .SIXBT  "DATAF0DAT"         /NAME OF DATA FILE USED BY PROC.
ERR3    .WRITE  TTO,ASC,MSG6,0      /FILE NOT FOUND
        JMP     CMULC+1             /ERASED FILE ? LOOK FOR NEXT FILE
MSG6    2000
        0
        .ASCII  "FILE NOT FOUND"<15>
AGN     LAC     AF1                 /INITIALIZE DATA-FILE
        DAC     FIL+1               / NAME
NDK3    SKP                         /DETERMINES # OF STORAGE DEVICES
        JMS     SWIT                /CHANGE STORAGE DEVICES
        JMP     DUM7+1              /NO, CONTINUE PROCESSING
ENDPR   .WRITE  TTO,ASC,END,0       /END OF PROC. MESSAGE
        .WAIT   TTO
        JMS     READM               /WAIT FOR RESPONSE AND PUT
ANSE    0                           /    THE ADDR. OF IT HERE
        LAC*    ANSE
        SAD     (120                /IS THE RESPONSE A "P" ?
        JMP     RSTR                /YES, SET UP TO RESTART EVERYTHING
        .IDLE                       /GIVE COMPLETE CONTROL TO BG.
RSTR    .CLEAR  DATIN               /CLEAR COLLECTION DEVICE
        .WRITE  TTO,ASC,MSG2,0      /DB MESSAGE
        .WAIT   TTO
        JMS     DBSUB               /SET UP NEXT DB SETTING
        JMS     READM               /WAIT FOR REPLY
DMRPLY  0                           /THE REPLY
        DZM     MCNTR               /ZERO BINARY TIMER
        .TIMER  0,BEGIN,6           /RESTART COLLECTION
        JMP     AGN                 /RESTART PROC.
/    SUBROUTINE USED TO WAIT FOR FILE TO BE COLLECTED AND STORED
/   WHILE GIVING TIME TO BACKGROUND
CONTR   0
WT2     LAC     NAME+1              /COLECTION'S FILE NAME
        SAD     FIL+1               /IS PROC.'S FILE NAME THE SAME ?
MDUM4   SKP                         /YES, GO TO WAIT
        JMP*    CONTR               /NO, RETURN TO PROCESS FILE
        ISZ     CTT11               /END OF THE
        SKP                         /    RUN ?
        JMP     ENDPR               /YES,EXIT
        .TIMER  30,WAIT1,0          /YES, RELINQUISH TIME TO BG.
        LAC     SAVAC               /RESTORE AC
        .IDLE                       /WAIT FOR CLOCK INTERRUPT
/REAL TIME SUB.--USED TO ALLOW TIME FOR BACKGROUND
WAIT1   0
        DAC     SAVAC               /SAVE AC
        DZM     WAIT1               /ZERO R-T SUB. ENTRY PT. TO ALLOW REENTR
        JMS     CKCOL               /CHECK FOR COLLECTION STOPPAGE
        JMP     WT2                 /CHECK FOR END OF COLL.
/     SUBROUTINE TO CHECK FOR UNWANTED COLL. STOPAGE
CKCOL   0
        LAW     -2
        TAD     CNT                 /HAS COLL. ENDED ALL COLLECTING
        SPA                         /    FOR TODAY?
        JMP*    CKCOL               /YES, RETURN
        LAC     TST1                /NO, HAS COLLECTION STOPPED
```

```
        TCA                         /    READING
        TAD     IDCOU               / IN DATA?
REPL    SZA                         /(REPLACED BY "NOP" WHEN COLL. IS DONE)
        JMP     SET1                /NO, RESET TESTER AND RETURN
        LAC     MDUM1               /YES, FREE TIMER
        DAC     ONCE                /    OPTION IN COLL.
        ISZ     CTT7                /TELLS IF STOPPAGE OCCURED
        .WRITE  TTO,ASC,STPM,0      /RING BELL
        .TIMER  0,CHECK,5           /RESTART COLLECTION

SET1    LAC     IDCOU               /SET ID # INTO
        DAC     TST1                /    TESTER FOR STOPPAGE
        JMP*    CKCOL               /CHECK FOR COLL. TO BE FINISH
/    SUB. TO CHANGE THE .DAT SLOT #'S TO CHANGE STORAGE DEVICES
SWIT    0
        LAC     (DATIN              /SWITCH STORAGE DEVICES
        ISZ     CTT4                /    BY CHANGING THE
        LAC     (DATIN2             /    .DAT SLOT IN COMMANDS:
        DAC     C11                 /    .SEEK
        DAC     C13                 /    .INIT
        DAC     C14                 /    .WAIT
        DAC     C15                 /    .CLOSE
        DAC     RSTR                /    .CLEAR
        TAD     (3000
        DAC     RT                  /    .FSTAT
        TAD     (1000
        DAC     LBA                 /    .READ
        LAC     CTT4                /IS DEVICE ON .DAT SLOT "DATIN"
        SMA                         /    TO BE PROCESSED?
        LAW     -2                  /YES, RESET DEVICE CONTROLLER
        DAC     CTT4                /MAKE ANY CHANGE IN CONTROLLER
        JMP*    SWIT
/
PP7     0
        LAS                         /GET THE CONSOLE DATA SWITCH
        AND     (40000              /    NUMBER 3
        SNA                         /IS IT A 1 ?
        JMS*    WRPP                /NO, PRINT DATA OUT ON PAPER TAPE
        JMP*    PP7                 /RETURN TO PROC
/
/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
        .EJECT
/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
/ .READ, DUMP MODE FROM DECTAPE ON A VARIABLE .DAT SLOT
/FILLS 252 DEC WORD BUFFER AND OUTPUTS 26
/WORDS TO ARRAY IDAT EVERY TIME CALLED.
/THESE ARE UNPACKED FROM 18 WORDS OF THE BUFFER.
/ IDAT: WORD  1         I.D. #
/       WORD 2-6        NOISE SAMPLES
/       WORD 7-27       DATA
/ NEGF:         SET IF A NEGATIVE NUMBER WAS IN THE DATA
/
DUMPT   0
        JMS*    .DA                 /PICKUP ADDR OF ADDR
        JMP     .+3                 /OF ARRAY
A2      0
FLAG    0                           /SET ON NEG #
        LAC*    A2                  /GET ADDR.
        DAC     A2                  /    OF ARRAY
        LAW     -NDPC               /SET COUNTER OF DATA TO BE
        DAC     COUNT               /    PROCESSED
        ISZ     CTT9                /GET POINTER
        JMP     LBB                 /NO, CONTINUE WITH PRESENT SET OF DATA
        LAW     -DATSTR

        DAC     CTT9
LBA     .READ   DATIN,DUMP,BUF3,DTBLK  /GET 1 BLOCK OF DATA
C14     .WAIT   DATIN
LCA     ISZ     SWITC               /INITIALLY READ TWO DUMMY BLOCKS
        JMP     LBA
        LAW     -3                  /RESET CONTROL TO READ
        DAC     SWITC               /    TWO DUMMY BLOCKS
        LAC     (JMP    LCB         /READ ONE BLOCK OF DATA
        DAC     LCA                 /    AT A TIME
LCB     LAC     BUF31               /GET ADDRESS OF BUF3
        DAC     POINT2              /    POINT2 TO BUF3
LBB     LAC*    POINT2              /GET THE ID (FIRST WORD IN DATA SET)
        SAD     SEVN                /END OF FILE ID?
        JMP     ENF                 /YES, RESET PARAM'S AND CLOSE FILE
        LAC*    POINT2              /GET ID AND PUT
        DAC*    A2                  /    INTO THE FORTRAN ARRAY
        ISZ     A2                  /GO TO NEXT ADDR. OF THE ARRAY
LOOP    ISZ     POINT2              /GO TO NEXT DATA WORD
        LAW     -2                  /PREPARE TO UNPAC
        DAC     TC10                /    TWO DATA WORDS
        LAC*    POINT2              /GET DATA WORDS FROM BUF3
```

```
        SWHA                            /FIRST WORD IN LEFT HALF
UNPLP   AND     (777                    /SAVE ONE DATA WORD
        SNA                             /CHECK FOR NEG. NUMBER
        ISZ*    FLAG                    /SET IF NEG. NUMBER FOUND
        DAC*    A2                      /LOAD # INTO FORTRAN ARRAY
        LAC*    POINT2                  /GET DATA WORD AGAIN
        ISZ     A2                      /GO TO NEXT LOC. IN ARRAY
        ISZ     TC10                    /UNPACED TWO WORDS?
        JMP     UNPLP                   /NO, LOOP AROUND
        ISZ     COUNT                   /YES, HAS 34 DATA WORDS BEEN UNPACKED?
        JMP     LOOP                    /NO, REPEAT UNPACKING PROCESS
OUT2    ISZ     POINT2                  /YES, GO TO NEXT ID
        JMP*    DUMPT                   /RETURN
SWITC   777775
POINT2  .DSA    BUF3
/-------------------------------------------
/ END OF FILE ROUTINE
ENF     LAC     SEVN                    /SET LAST ID TO
        DAC*    A2                      /     130050 DECIMAL
C15     .CLOSE  DATIN                   /CLOSE FILE
RT2     JMP*    DUMPT                   /RETURN TO PROC. PROGRAM
/
/ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
        .END    INTIM
```

```
CCCCCCCCCCCCCCCCCCCCC--CONTL--CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       PROGRAM SETS UP CALIBRATION AND THE HEADING FOR THE PRINT
C OUT OF THE PROCESSED DATA IN PROC.  THE PROGRAM CALLS:
C               TBFORL--CALIBRATION PROGRAM
C               DLOGF---INITIAL MACRO PROGRAM
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
        SUBROUTINE CONTL(ISURP)
        INTEGER DB(4),DBC
        REAL DATE(2),REAS(5)
        COMMON /STAT/ DB,DATE,REAS,DBC,DBS,NC4
        DATA NR/120784/
        REWIND 4
C  INITIALIZE THE DB SETTINGS USED
        DBC=3
        DB(2)=10
        DB(3)=25
        IF(ISURP.NE.0)GO TO 5
C   GIVE PRECALIBRATION SETUP
        WRITE(6,100)
100     FORMAT(46H TURN OFF PULSER AND ENCODE PULSE POWER SUPPLY/)
C  SET UP CALIBRATION AND LINEARIZATION TABLE
        CALL TBFORL
C  ASK FOR AND GET THE DATE
5       WRITE(6,101)NR
101     FORMAT(5H DATE,A2)
        READ(4,201)DATE
201     FORMAT (2A5)
C  ASK FOR AND GET THE REASON FOR THE RUN
        WRITE(6,102)NR
102     FORMAT(16H REASON FOR DATA,A2)
        READ(4,202)REAS
202     FORMAT(5A5)
C  PREPARE FOR COLLECTION AND PROCESSING
        CALL DLOGF(DB,DBC,DATE)
        RETURN
        END
```

```
CTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT---TBFORL---TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
C       THIS PROGRAM SETS UP A LINEAR APPROXIMATION TO THE INPUT OF
C THE RECEIVER VERSUS THE OUTPUT OF THE A/D CONVERTER AS READ BY THE
C COMPUTER.  FROM THISLINEAR APPROXIMATION, A TABLE IS FORMED BY
C USING INPUTS FROM .5 TO 511.5 INCREMENTED BY 1.  THE OFFSET OF .5
C IS USED FOR BETTER ACCURACY IN THE ROUNDOFF ERROR OF THE A/D CONVER-
C TER.  ALL THE VALUES OF ZERO IN THE TABLE ARE CHANGED TO 1 SINCE
C 0 IS USED TO DESIGNATE NEGATIVE NUMBERS DURING THE COLLECTION.
C THE PROGRAM CALLS THE SUBROUTINES:
C               RADC---READS #'S FROM THE A/D CONVERTER (MACRO)
C               LINAP--CONVERTS INPUTS TO OUTPUTS FOR THE FORMATION
C                       OF THE LINEARIZATION TABLE
C               TTM----WRITES LIN. TABLE ONTO A STORAGE DEVICE
CTTTTTTTTT TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
```

```
C
      SUBROUTINE TBFORL
      INTEGER START(512),DLPO,STAT,IAS(44)
      REAL KI
      COMMON /TA/ S(43),TU(44),TUO(44)
      DATA C,CS,CN,NR/1HC,1HS,1HN,120784/
C   DETERMINE IF CALIBRATION IS NEEDED AND WHICH PRINTOUT TO USE
7     WRITE(6,105)NR
105   FORMAT(18H WHICH CALIBRATION/24H (S-SHORT, R-REGULAR, OR
     137H C-COMPLETE PRINT OUT OR N-NO CALIB.),A2)
      READ(4,204)CAL
204   FORMAT(A1)
      IERR=0
      DLPO=0
C   SET UP THE WANTED CALIBRATION
      IF(CAL.EQ.CN)RETURN
      IF(CAL.EQ.CS)DLPO=-1
      IF(CAL.EQ.C)DLPO=1
C   NUMBER OF A/D NUMBERS TO READ PER DB SETTING=NAV1*NAV
      NAV1=5
      NAV=500
      AV=NAV1*NAV
      ICT=0
C   NUMBER OF ATTENUATOR SETTINGS=NT1
      NT1=42
      NT2=NT1-1
      NT3=NT1-2
C   INPUT SIGNALS--FROM 5DB (OF 1000) TO INFINITY
      TU(NT1)=562.34
      TU(NT2)=501.19
      TU(NT3)=446.68
      TU(NT1-3)=398.11
      TU(NT1-4)=354.82
      TU(NT1-5)=316.23
      TU(NT1-6)=281.84
      TU(NT1-7)=251.19
      TU(NT1-8)=223.87
      TU(NT1-9)=199.53
      TU(NT1-10)=177.83
      TU(NT1-11)=158.49
      TU(NT1-12)=141.25
      TU(NT1-13)=125.90
      TU(NT1-14)=112.20
      TU(NT1-15)=100.00
      TU(NT1-16)= 89.13
      TU(NT1-17)= 79.43
      TU(NT1-18)= 70.80
      TU(NT1-19)= 63.10
      TU(NT1-20)= 56.24
      TU(NT1-21)= 50.12
      TU(NT1-22)= 44.67
      TU(NT1-23)= 39.81
      TU(NT1-24)= 35.48
      TU(NT1-25)= 31.62
      TU(NT1-26)= 28.18
      TU(NT1-27)= 25.12
      TU(NT1-28)= 22.39
      TU(NT1-29)= 19.95
      TU(NT1-30)= 17.78
      TU(NT1-31)= 15.85
      TU(NT1-32)= 14.13
      TU(NT1-33)= 12.59
      TU(NT1-34)= 11.22
      TU(NT1-35)= 10.00
      TU(NT1-36)=  8.91
      TU(NT1-37)=  7.94
      TU(NT1-38)=  7.08
      TU(NT1-39)=  6.31
      TU(NT1-40)=  5.62
      TU(1)=  0.00
20    TUO(NT1)=512.
C   SET UP MESSAGES FOR TELLING WHICH ATTENUATOR SETTING TO DO
      DO 11 I=1,NT3
11    IAS(I)=I+5
      IAS(NT2)=99
C   DO LOOP TO INPUT ALL THE OUTPUTED SIGNALS
      DO 2 I1=1,NT2
      I2=NT1-I1
      DUM=0
      WRITE (6,100) IAS(I1),NR
100   FORMAT(8H SET TO ,I2,23HDB ATTENUATION AND C.R.,A2)
      READ(4,200)F
200   FORMAT(F5.1)
C   IF THE INPUTED NUMBER IS TWO DIGITS RESTART THE SETTINGS
      IF(F.GT.10.)GO TO 20
```

```
C     ISSUE AN A/D CONVERTER READ "NAV1" TIMES
          DO 10 J1=1,NAV1
C     READ "NAV" NUMBERS FROM THE A/D CONVERTER
          CALL RADC(START,NAV,STAT)
400       IF(STAT.EQ.0)GO TO 400
          DO 1 J=1,NAV
          IF(START(J).GT.511) START(J)=0
C     STORE THE INPUTED NUMBERS IN A REAL VARIABLE
1         DUM= DUM+FLOAT(START(J))

10        CONTINUE
C     AVERAGE THE OUTPUTED NUMBERS AND GO TO THE NEXT SETTING
2         TUO(I2)=DUM/AV
          DO 3 I1=1,NT2
          I2=I1+1
          ICT=ICT+1
          I3=NT1-I1
C     SET UP THE SLOPES OF EACH LINE SEGMENT APPROXIMATION
          S(I1)=(TU(I2)-TU(I1))/(TUO(I2)-TUO(I1))
          IF(ICT.EQ.2)GO TO 3
C     WRITE OUT STRAIGHT LINE APPROXIMATION TABLE
          WRITE(6,102) I1,S(I1),I1,TU(I1),I1,TUO(I1),IAS(I3)
          ICT=1
          IF(DLPO.GT.-1)ICT=0
C     POSSIBLE ERROR CONDITIONS FOR THE APPROXIMATION JUST FORMED
3         IF(S(I1).LT..3.OR.S(I1).GT.7.) IERR=IERR+1
102       FORMAT (4H  S(,I2,2H)=,F6.3,5X,3HTU(,I2,2H)=,F8.3,5X,
          14HTUO(,I2,2H)=,F8.3,3X,I2,2HDB)
C     WRITE OUT LAST VALUES OF THE TABLE
          WRITE(6,101) TU(NT1),TUO(NT1)
101       FORMAT(19X,7HTU(42)=,F8.3,5X,8HTUO(42)=,F8.3,7H     5DB///)
          IERR2=0
C     FINAL OUTPUT VALUE FOR THE LINEARIZATION TABLE FORMATION
          XF=511.5
C     GET THE INPUT VALUE AND STORE IN "XF"
          CALL LINAP(XF,NT1)
          IF (XF.LE.0.)XF=512
C     NORMALIZATION FACTOR OF THE OUTPUT VALUES OF THE LIN. TABLE
          K1=1023./XF
C     INITIAL OUTPUT USED TO DETERMINE THE INPUT VALUE
          X1=511.5/1024.
C     FIRST INPUT VALUE OF THE TABLE
          START (1) =1
          DO 4 I=2,512
C     NEXT OUT PUT VALUE USED
          X3=X1*FLOAT(2*I-1)
C     GET THE INPUT VALUE AND STORE IN "X3"
          CALL LINAP (X3,NT1)
          X3=(X3*K1+1.)/2.
          IF(X3.LE.1.)X3=1.1
C     STORE INPUT VALUE IN INTEGER LIN. TABLE
          START(I)=IFIX(X3)
C     ERROR CONDITION FOR LINEARIZATION TABLE
4         IF(START(I)+1.LT.START(I-1))IERR2=IERR2+1
C     LAST VALUE OF THE TABLE
          START(512)=511
          IF(DLPO.LT.1) GO TO 6
C     NEW PAGE
          WRITE (6,103)
103       FORMAT (1H1)
C     WRITE LINEARIZATION TABLE ON TELETYPE
          WRITE (6,104)  (START(I),I=1,512)
104       FORMAT (10(I5,2X))
C     NEW PAGE

          WRITE(6,103)
C     WRITE TABLE ON A STORAGE DEVIVE (DUMP MODE)
6         CALL TTM(START)
C     WRITE OUT ANY ERROR AND ALLOW RECALIBRATION IF NEEDED
          IF(IERR2.NE.0)GO TO 9
          IF(IERR.EQ.0)RETURN
          WRITE(6,106)
106       FORMAT(//47H  ****CHECK CALIBRATION FOR POSSIBLE ERRORS****///)
          GO TO 7
9         WRITE(6,107)
107       FORMAT(37H +++ ERROR--BAD CALIBRATION TABLE +++)
          GO TO 7
          RETURN
          END
```

```
C        *********************SUBROUTINE LINAP*********************
C        LINAP TRANSFORMS OUTPUT VOLTAGES INTO INPUT VOLTAGES OF
C        THE RECEIVER. THE CALIBRATION DATA IS CONTAINED IN SUB-
C        ROUTINE VALUE.
C        *******************************************************
C
C        INPUT AND OUTPUT:
C        A IS THE OUTPUT VOLTAGE THAT IS TRANSFORMED INTO
C        INPUT  VOLTAGE
C        NC5 IS THE NUMBER OF DB SETTINGS
C
         SUBROUTINE LINAP(A,NC5)
         COMMON /TA/ S(43),TU(44),TUO(44)
         N=1
C    DIVIDE THE STRAIGHT LINE APPROXIMATION INTO 4 AREAS
         NC54=(NC5+2)/4
         NC53=NC5*3/4
         NC52=(NC5+1)/2
C    FIND WHERE THE INPUTED NUMBER LIES
         IF(A.GT.TUO(NC54))N=NC54
         IF(A.GT.TUO(NC52))N=NC52
         IF(A.GT.TUO(NC53))N=NC53
C    SET THE UPPER LIMIT OF THE SEARCH
         K=N+NC54
C    SEARCH FOR THE CORRECT LINE SEGMENT
         DO 5 I=N,K
         J=I+1
         IF(A.GT.TUO(I).AND.A.LE.TUO(J)) GO TO 10
5        CONTINUE
C    LINE SEGMENT COULDN'T BE FOUND
         A=0.
         RETURN
C    GET THE VALUE OF THE CORRESPONDING OUTPUT
10       A=(A-TUO(I))*S(I)+TU(I)
         RETURN
         END
```

```
CPPPPPPPPPPPPPPPPPPPPPPPPPPP--PROC--PPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
C        THIS PROGRAM OPERATES ALONG WITH DLOGF1.  IT TAKES DATA READ
C BY A SUBROUTINE AND CHECKS THE NOISE AND SATURATIONS
C FOR GOOD DATA.  IT SUMS THE SQUARES OF THE DATA AT EACH HEIGHT
C AND THE ACCEPTABLE NOISE, SUBTRACTS THE ACCEPTABLE NOISE SQUARED
C FROM THDATA, TAKES THE SQUARE ROOT OF THE DATA PRESERVING THE
C SIGN, TAKES THE SQUARE ROOT OF THE NOISE, AND PRINTS OUT THE RESULTS
C ON THE TELETYPE.  THE RESULTS MAY ALSO BE PRINTED OUT ON PAPER TAPE
C IF NEEDED.  THE PROGRAMS USED BY PROC ARE:
C                  CHNG----INITIALIZES STORAGE DEVISE AND WAITS FOR FILE
C                          TO BE COLLECTED PLUS COMMUNICATING WITH COLL.
C                  DRD73--STORES DATA IN REAL ARRAY (F0RTRAN)
C                  CALC2---CALCULATES ELECTRON DENSITIES (FORTRAN)
C                  CKCOL---CHECKS FOR UNWANTED STOPS IN COLLECTION
C                          PROGRAM (MACRO)
C                  PP7-----CHECKS DATA SWICHES ON THE CONSOLE TO ALLOW
C                          OR DISALLOW RESULTS TO BE PUT ON PAPER TAPE
CPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
C
         SUBROUTINE PROC(IA,PLF,ITIME)
         COMMON /PPC/ AO(21),AX(21),AVAO(21),AVAX(21),ITIM(4),
         1XO(21),IRJ(21),BNO(5),BNX(5),EL(21),
         2RBMX,AVNO,BMO,AVNX,BMX,ID,IR
         COMMON /STAT/ IDB(4),RDATE(2),REAS(5),IDBC,BMXNS,NC4
         SRBMX=3.1
         RBMX=SRBMX*SRBMX
         IWIN=-1
         IEQU=0
         ISUM=1
         NC4=1
C   INITIALIZE PAPER PUNCH
         WRITE(7,1500)
C  INITIALIZE ALL VARIABLES NEEDED TO BE INIT.
10       SNO=0.
         SNX=0.
         IR=0
         IRN=0
         BMO=0.
         BMX=0.
         DO 100 I=1,21
         AVAO(I)=0.
         AVAX(I)=0.
         AO(I)=0.
         AX(I)=0.
100      IRJ(I)=0
```

```
      DO 110 I=1,4
      BNO(I)=0.
110   BNX(I)=0.
C  INITIALIZE STORAGE DEVICE AND
C  PREPARE TO READ DATA
      CALL CHNG(IBMX,IDBCH,IMCC)
      IF(IMCC.LE.0.OR.IMCC.GT.9)GO TO 80
      CMC=FLOAT(IMCC)-4.5
      RBMX=RBMX+CMC*CMC*.4
20    IF(IBMX.LT.1)IBMX=216
      IMCC=0
C             NOISE CRITERION
C   IBMX=THE SUM OF THE SET OF 45 NOISE SAMPLES WHICH HAS THE
C  MIN. MAXIMUM OF ALL THE MAXIMUMS OF EACH SET OF 45 NOISE SAMPLES
C   THE AVERAGE NOISE FOR THE FIRST 5 IN EACH FRAME HAS
C  TO BE LESS THAN SRBMX*IBMX/45.  WHERE RBMX IS THE SUPPLIED CONSTANT.
C  FOR SPEED RBMX*(IBMX/45.)**2*5. IS COMPARED TO THE SUM OF THE
C  SQUARES OF THE NOISE.
      DUMX=FLOAT(IBMX)/45.
      IF(DUMX*SRBMX.GT.500.)DUMX=500./SRBMX
      BMXNS=RBMX*DUMX*5.*DUMX
      KEOFO=0
      KEOFX=0
      ID=0
C   GET ORDINARY MODE DATA
30    CALL DRD73(AO,BNO,IERR,ID,KEOFO)
      IF(KEOFO.EQ.1)GO TO 80
C   GET EXTRAORDINARY MODE DATA
      CALL DRD73(AX,BNX,IERR,ID,KEOFX)
      IF(KEOFX.EQ.1)GO TO 80
C   SET UP CHECK FOR REJECTION BECAUSE OF NOISE CRITERION
      BMEANO=0.
      BMEANX=0.
      DO 120 I=1,5
      BMEANO=BMEANO+BNO(I)*BNO(I)
120   BMEANX=BMEANX+BNX(I)*BNX(I)
      IF(BMEANO.GT.BMXNS.OR.BMEANX.GT.BMXNS)GO TO 50
C   NOISE USED TO SUTRACT FROM DATA SAMPLES
      BMO=BMO+BMEANO
      BMX=BMX+BMEANX
C   SUM OF THE SQUARE OF THE UNSATURATED DATA AT EACH HEIGHT
      DO 140 I=1,21
35    IF(AO(I).GE.510..OR.AX(I).GE.510.)GO TO 40
      AVAO(I)=AVAO(I)+AO(I)*AO(I)
      AVAX(I)=AVAX(I)+AX(I)*AX(I)
      GO TO 140
C   REJECTIONS DUE TO SATURATIONS OF DATA
40    IRJ(I)=IRJ(I)+1
140   CONTINUE
      GO TO 60
C   REJECTIONS FROM NOISE CRITERION
50    IR=IR+1
C   SET UP AVERAGE NOISE USED IN REJECTION CRITERION
60    SNO=BMEANO+SNO
      SNX=BMEANX+SNX
      GO TO 30
80    ID=ID/2
      BID=ID*5
C   MAXIMUM ALLOWABLE NOISE
      BMXNS=BMXNS/(DUMX*5.*SRBMX)
C   RMS OF ALL NOISE SAMPLES
      AVNO=SQRT(SNO/BID)
      AVNX=SQRT(SNX/BID)
C   NUMBER OF ACCEPTABLE NOISE SAMPLES
      RN=5*(ID-IR)
      DO 150 I=1,21
C   NUMBER OF REJECTIONS AT EACH HEIGHT
      IRJ(I)=IRJ(I)+IR
C   NUMBER OF ACCEPTABLE DATA AT EACH HEIGHT
      RSAM=ID-IRJ(I)
C   AVERAGE SUM SQUARED OF ACCEPTABLE DATA FOR EACH HEIGHT MINUS THE
C   AVERAGE SUM SQUARED OF THE ACCEPTABLE NOISE
      AVOC=AVAO(I)/RSAM-BMO/RN
      AVXC=AVAX(I)/RSAM-BMX/RN
C   THE RMS OF THE ACCEPTABLE DATA AT EACH HEIGHT (PRESERVING THE SIGN)
      AVAO(I)=(ABS(AVOC)/AVOC)*SQRT(ABS(AVOC))
      AVAX(I)=(ABS(AVXC)/AVXC)*SQRT(ABS(AVXC))
      EL(I)=0.
      XO(I)=0.
      IF(AVAO(I).LE.0.0.OR.AVAX(I).LE.0.0)GO TO 150
      XO(I)=AVAX(I)/AVAO(I)
```

```
150       CONTINUE
C    THE RMS OF THE ACCEPTABLE NOISE
          BMO=SQRT(BMO/RN)
          BMX=SQRT(BMX/RN)
          CALL CALC2(XO,1,20,EL,IA)
C    GET THE TIME OF DAY
          DO 155 I=1,4
          I1=5-I
155       ITIM(I)=ITIME/(10**I1)
C    WRITE THE HEADING ON THE TELETYPE
          WRITE(6,1050)ITIM,RDATE,REAS,IDB(NC4)
1050      FORMAT(1H1,4I1,4X,2A5,3X,5A5,3X,I2,2HDB)
          WRITE (6,1100) BMXNS,RBMX
1100      FORMAT(//19H MAX. ALLOW. NOISE=,F7.1,16H   MULT. CONST.=,F7.3/)
          WRITE(6,1200)AVNO,BMO,AVNX,BMX
1200      FORMAT(16H O-NOISE AV.(1) ,F8.1,7H   (2) ,F8.1/
          116H X-NOISE AV.(1) ,F8.1,7H   (2) ,F8.1)
          WRITE(6,1300)ID,IR
1300      FORMAT(//1X,I4,8H SAMPLES,5X,I5,12H REJ.(NOISE)//3X,
          14HREJ./48H (N.+SAT.)   HEIGHT   AV. AO  AV. AX  AX/AO   ED/)
          HT=5 8.5
          DO 160 I=1,21
C    CHECKS FOR COLLECTION STOPPAGE
          CALL CKCOL
          HT=HT+1.5
          WRITE(6,1400)IRJ(I),HT,AVAO(I),AVAX(I),XO(I),PLF,EL(I)
1400      FORMAT(4X,I4,4X,F5.1,3X,F6.1,2X,F6.1,2X,F5.2,A3,F6.0)
160       CONTINUE
C    ALLOWS RESULTS TO BE SAVED ON PAPER TAPE
          CALL PP7
C    NEXT ATTENUATOR SETTING
          NC4=NC4+1
          IF(NC4.GT.IDBC)NC4=1
          IF(IDBCH.GT.0)IDB(NC4)=5*IDBCH
          GO TO 10
1500      FORMAT(1H )
          RETURN
          END




C************************SUBROUTINE DRD73**************************
C         DREAD READS 21 SAMPLES OF SIGNAL AND 5 SAMPLES OF NOISE
C         FROM DECTAPE. THE OUTPUT VOLTAGES HAVE BEEN TRANSFORMED INTO
C         INPUT VOLTAGES.  THE PROGRAM USES SUBROUTINE DUMPT (MACRO) TO
C         READ DATA FROM STORAGE DEVICE.
C*****************************************************************
C
          SUBROUTINE DRD73(A,BMEAN,IERR,ID,KEOF)
          DIMENSION A(21),IDAT(27),BMEAN(5)
          KEOF=0
          N=5
          N1=N+1
          N2=N+2
          N3=N1+21
C    GET ONE SET OF DATA (26 NUMBERS)
          CALL DUMPT(IDAT,NEGF)
C CHECK ID CONSECUTIVE
          IF(ID-IDAT(1)+1) 10,15,10
C     CHECK FOR END OF FILE
10        IF(IDAT(1).NE.130050) GO TO 20
C    TELL PROC IT'S THE END OF THE FILE
        . KEOF=1
          RETURN
15        ID=IDAT(1)
C    SET DATA SAMPLES INTO A REAL ARRAY
40        DO 42 MIN=N2,N3
          MFVE=MIN-N1
          A(MFVE)=IDAT(MIN)
42        CONTINUE
C    SET NOISE SAMPLES INTO A REAL NUMBER ARRAY
          DO 130 J=1,N
          JEL=J+1
130       BMEAN(J)=IDAT(JEL)
          RETURN
C    THE ID IS ERRONEOUS, IGNORE THE REST OF THE DATA
20        WRITE(6,100)ID,IDAT(1),IDAT(2)
100       FORMAT(44H ID WAS NOT CONSECUTIVE AND NOT=130050 ; ID=,3(I7,3X))
          KEOF=1
          RETURN
          END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCC---CALC2---CCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       CALC2 IS A LIST OF CONSTANTS  CALUATED FROM THE PROGRAMS
C CALC AND ELDEN AND CONTAINS THE FUNCTION THAT CALCULATES THE
C ELECTRON DENSITIES FOR THE PARTIAL-REFLECTION PROCESSING PROGRAMS.
C THE PROGRAM CALC WRITES THIS PROGRAM.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE CALC2(ARRAY,LL,LH,FD,IA)
        DIMENSION ARRAY(21),RATIO2(21),FD(21)
C   GET THE PREDETERMINED CONSTANTS FOR THE RIGHT SEASON
        IF(IA)200,300,100
C  CONSTANTS FOR THE SUMMER
100     RATIO2( 1)= 1.0731152
        RATIO2( 2)= 1.0778633
        RATIO2( 3)= 1.0841143
        RATIO2( 4)= 1.0909986
        RATIO2( 5)= 1.0990518
        RATIO2( 6)= 1.0901243
        RATIO2( 7)= 1.0880302
        RATIO2( 8)= 1.0864129
        RATIO2( 9)= 1.0768397
        RATIO2(10)= 1.0590323
        RATIO2(11)= 1.0495014
        RATIO2(12)= 1.0328141
        RATIO2(13)= 1.0262468
        RATIO2(14)= 1.0166022
        RATIO2(15)= 1.0117923
        RATIO2(16)= 1.0076428
        RATIO2(17)= 1.0044388
        RATIO2(18)= 1.0029128
        RATIO2(19)= 1.0015084
        RATIO2(20)= 1.0008443
        FD( 1)= 0.170327E-03
        FD( 2)= 0.243443E-03
        FD( 3)= 0.329813E-03
        FD( 4)= 0.427651E-03
        FD( 5)= 0.532098E-03
        FD( 6)= 0.627513E-03
        FD( 7)= 0.699730E-03
        FD( 8)= 0.744879E-03
        FD( 9)= 0.753246E-03
        FD(10)= 0.722645E-03
        FD(11)= 0.660879E-03
        FD(12)= 0.579908E-03
        FD(13)= 0.492934E-03
        FD(14)= 0.405772E-03
        FD(15)= 0.328260E-03
        FD(16)= 0.258602E-03
        FD(17)= 0.201748E-03
        FD(18)= 0.155429E-03
        FD(19)= 0.118415E-03
        FD(20)= 0.915812E-04
        GO TO 400
C  CONSTANTS FOR THE WINTER
200     RATIO2( 1)= 1.0682487

        RATIO2( 2)= 1.0890572
        RATIO2( 3)= 1.0699695
        RATIO2( 4)= 1.0979884
        RATIO2( 5)= 1.0692204
        RATIO2( 6)= 1.0854059
        RATIO2( 7)= 1.0770541
        RATIO2( 8)= 1.0745673
        RATIO2( 9)= 1.0665843
        RATIO2(10)= 1.0531266
        RATIO2(11)= 1.0458097
        RATIO2(12)= 1.0329590
        RATIO2(13)= 1.0286678
        RATIO2(14)= 1.0167225
        RATIO2(15)= 1.0127123
        RATIO2(16)= 1.0076428
        RATIO2(17)= 1.0046818
        RATIO2(18)= 1.0023713
        RATIO2(19)= 1.0019426
        RATIO2(20)= 1.0007090
        FD( 1)= 0.237897E-03
        FD( 2)= 0.319960E-03
        FD( 3)= 0.410827E-03
        FD( 4)= 0.502861E-03
        FD( 5)= 0.592594E-03
        FD( 6)= 0.662825E-03
        FD( 7)= 0.720342E-03
        FD( 8)= 0.750909E-03
        FD( 9)= 0.751034E-03
        FD(10)= 0.720293E-03
        FD(11)= 0.663644E-03
```

```
      FD(12)= 0.588398E-03
      FD(13)= 0.500527E-03
      FD(14)= 0.410427E-03
      FD(15)= 0.330916E-03
      FD(16)= 0.258602E-03
      FD(17)= 0.200210E-03
      FD(18)= 0.156383E-03
      FD(19)= 0.119390E-03
      FD(20)= 0.899351E-04
      GO TO 400
C  CONSTANTS FOR EQUINOX
300   RATIO2( 1)= 1.0670392
      RATIO2( 2)= 1.0803050
      RATIO2( 3)= 1.0780858
      RATIO2( 4)= 1.0912947
      RATIO2( 5)= 1.0914300
      RATIO2( 6)= 1.0917337
      RATIO2( 7)= 1.0897206
      RATIO2( 8)= 1.0825763
      RATIO2( 9)= 1.0714581
      RATIO2(10)= 1.0590323
      RATIO2(11)= 1.0482702
      RATIO2(12)= 1.0351681
      RATIO2(13)= 1.0240435
      RATIO2(14)= 1.0176469
      RATIO2(15)= 1.0117923
      RATIO2(16)= 1.0070148
      RATIO2(17)= 1.0048179
      RATIO2(18)= 1.0027836
      RATIO2(19)= 1.0015986
      RATIO2(20)= 1.0007970
      FD( 1)= 0.188515E-03
      FD( 2)= 0.261225E-03
      FD( 3)= 0.347697E-03
      FD( 4)= 0.442169E-03
      FD( 5)= 0.542672E-03
      FD( 6)= 0.633402E-03
      FD( 7)= 0.705094E-03
      FD( 8)= 0.747132E-03
      FD( 9)= 0.752781E-03
      FD(10)= 0.722645E-03
      FD(11)= 0.661807E-03
      FD(12)= 0.579908E-03
      FD(13)= 0.492934E-03
      FD(14)= 0.408105E-03
      FD(15)= 0.328260E-03
      FD(16)= 0.261505E-03
      FD(17)= 0.206349E-03
      FD(18)= 0.160195E-03
      FD(19)= 0.124908E-03
      FD(20)= 0.997959E-04
400   DO 10 I=LL,LH
      IF(ARRAY(I).EQ.0..OR.ARRAY(I+1).EQ.0.)GO TO 50
C   THE FUNCTION FOR THE CALCULATION OF ELECTRON DENSITIES
      FD(I)=ALOG((ARRAY(I)/ARRAY(I+1))*RATIO2(I))/FD(I)
      GO TO 10
50    FD(I)=0.
10    CONTINUE
      RETURN



CPPPPPPPPPPPPPPPPPPPPPPPPP---WRPP---PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
C  WRPP PUNCHES THE PROCESSED PARTIAL REFLECTION DATA
C  ON PAPER TAPE
CPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
C
      SUBROUTINE WRPP
      COMMON /PPC/ AO(21),AX(21),AVAO(21),AVAX(21),ITIM(4),
     1XO(21),IRJ(21),BNO(5),BNX(5),EL(21),
     2RBMX,AVNO,BMO,AVNX,BMX,ID,IR
      COMMON /STAT/ IDB(4),RDATE(2),REAS(5),IDBC,BMXNS,NC4
      WRITE(7,1050)ITIM,RDATE,REAS,IDB(NC4),BMXNS,RBMX,
     1AVNO,BMO,AVNX,BMX,ID,IR
1050  FORMAT(1H1,4I1,4X,2A5,3X,5A5,3X,I2,2HDB/F7.1,F7.3,F8.1,
     1F8.1,F8.1,F8.1,I4,I5)
      DO 160 I=1,21
      CALL CKCOL
      WRITE(7,1400)IRJ(I),AVAO(I),AVAX(I),XO(I),EL(I)
1400  FORMAT(1X,I4,F6.1,F6.1,F5.2,F6.0)
160   CONTINUE
      RETURN
      END
```

```
C***********************-PROC73-**************************************
C       PROC73 EVALUATES COLLECTED PARTIAL-REFLECTION DATA AND
C PRINTS OUT THE ELECTRON DENSITY.  PROC73 USES THE FOLLOWING PROGRAMS:
C               HEAD----SETS UP AND PRINTS THE HEADING (FORTRAN)
C               DINIT---INITIALIZES THE STORAGE DEVIVE (MACRO)
C               FSTAT---LOCATES THE DATA FILE (MACRO)
C               SEEK----FINDS THE FILE ON THE STORAGE DEVICE (MACRO)
C               DRD73--SETSSAMPLES INTO THE REAL # ARRAY (FORTRAN)
C               CALC2---CALCULATES THE ELECTRON DENSITY (FORTRAN)
C********************************************************************
C
        INTEGER DATIN
        DIMENSION FNAM(2),AO(21),AX(21),AVAO(21),AVAX(21),
        1XO(21),IRJ(21),BNO(5),BNX(5),EL(21)
        WRITE(6,105)
105     FORMAT(48H TYPE IN SEASON--(1) FOR SUMMER, (-1) FOR WINTER
        115H, (0) OTHERWISE)
        READ(4,200)IA
200     FORMAT(I2)
        DATIN=2
10      CALL HEAD(0)
C   INITIALIZE VARIABLES
12      SNO=0.
        SNX=0.
        IR=0
        IRN=0
        BMO=0.
        BMX=0.
        DO 16 I=1,21
        AVAO(I)=0.
        AVAX(I)=0.
16      IRJ(I)=0
        DO 17 I=1,4
        BNO(I)=0.
17      BNX(I)=0.
C   INITIALIZE TAPE STORING THE DATA
        CALL DINIT
C   GET THE DATA FILE NAME
        WRITE(6,20)
20      FORMAT(15H WHICH DATAFILE)
        READ(4,30)FNAM
30      FORMAT(2A5)
C   CHECK THE VALIDITY OF THE NAME GIVEN
13      CALL FSTAT(DATIN,FNAM,LOG)
        IF(LOG.NE.0)GO TO 40
        WRITE(6,35)FNAM
35      FORMAT(6H FILE ,2A5,19H NOT FOUND ON DAT 2)
        GO TO 10
C   FIND LOCATION OF FILE ON THE TAPE
40      CALL SEEK(DATIN,FNAM)
C   GET THE MAXIMUM ALLOWABLE NOISE
        WRITE(6,57)
57      FORMAT(14H MAXIMUM NOISE)
        READ(4,56)BMXNS
56      FORMAT(F10.0)
        IF(BMXNS.GE.510.)BMXNS=400.
C   FOR SPEED USE THE SQUARE OF THE MAX. ALLOW. NOISE TIMES 5
        DUM4=BMXNS*5.
        BMXNS=BMXNS*DUM4
19      KEOFO=0
        KEOFX=0
        ID=0
C   GET ONE SET OF 26 ORDINARY SAMPLES
48      CALL DRD73(AO,BNO,IERR,ID,KEOFO)
        IF(KEOFO.EQ.1)GO TO 50
C   GET ONE SET OF 26 EXTRAORDINARY SAMPLES
        CALL DRD73(AX,BNX,IERR,ID,KEOFX)
        IF(KEOFX.EQ.1)GO TO 49
C   GET THE SUM SQUARED OF THE NOISE
        BMEANO=0.
        BMEANX=0.
        DO 440 I=1,5
        BMEANO=BMEANO+BNO(I)*BNO(I)
440     BMEANX=BMEANX+BNX(I)*BNX(I)
C   CHECK FOR SETS OF SAMPLES THAT ARE TOO NOISY
        IF(BMEANO.GT.BMXNS.OR.BMEANX.GT.BMXNS)GO TO 510
C  SUM THE SQUARED NOISE SAMPLES FOR THE
C    LAST FOUR NOISE SAMPLES PER 25 TOTAL SAMPLES
        BMO=BMO+BMEANO-BNO(1)*BNO(1)
        BMX=BMX+BMEANX-BNX(1)*BNX(1)
730     DO 47 I=1,21
        IF(AO(I).GE.510..OR.AX(I).GE.510.)GO TO 46
```

```
C  SUM OF THE SQUARED GOOD AO AND AX SAMPLES
       AVAO(I)=AVAO(I)+AO(I)*AO(I)
120    AVAX(I)=AVAX(I)+AX(I)*AX(I)
       GO TO 47
C  THE TOTAL # OF REJECTIONS DUE TO SATURATION
C    PLUS NOISE ABOVE THE GIVEN MAXIMUM
46     IRJ(I)=IRJ(I)+1
47     CONTINUE
       GO TO 520
510    IR=IR+1
520    SNO=BMEANO+SNO
       SNX=BMEANX+SNX
       GO TO 48
49     ID=ID-1
50     ID=ID/2
       BID=ID*5
       BMXNS=BMXNS/DUM4
C   THE RMS OF ALL NOISE SAMPLES TAKEN
       AVNO=SQRT(SNO/BID)
       AVNX=SQRT(SNX/BID)
C   THE NUMBER OF THE ACCEPTABLE NOISE SAMPLES
       RN=4*(ID-IR)
       DO 52 I=1,21
C   NUMBER OF REJECTED DATA AT EACH HEIGHT
       IRJ(I)=IRJ(I)+IR
C   NUMBER OF ACCEPTABLE DATA AT EACH HEIGHT
       RSAM=ID-IRJ(I)
C   ACCEPTABLE NOISE IS SUBTRACTED OFF
       AVOC=AVAO(I)/RSAM-BMO/RN
       AVXC=AVAX(I)/RSAM-BMX/RN
C   RMS OF GOOD DATA WITH THE SIGN PRESERVED
       AVAO(I)=(ABS(AVOC)/AVOC)*SQRT(ABS(AVOC))
       AVAX(I)=(ABS(AVXC)/AVXC)*SQRT(ABS(AVXC))
       EL(I)=0
       XO(I)=0
       IF(AVAO(I).LE.0.0.OR.AVAX(I).LE.0.0)GO TO 52
       XO(I)=AVAX(I)/AVAO(I)
52     CONTINUE
C   RMS OF ACCEPTABLE NOISE
       BMO=SQRT(BMO/RN)
       BMX=SQRT(BMX/RN)
C   GET ELECTRON DENSITIES
       CALL CALC2(XO,1,20,EL,IA)
C   WRITE THE HEADING
       CALL HEAD(1)
       WRITE (6,100) BMXNS,AVNO,BMO,AVNX,BMX
100    FORMAT (25H MAXIMUM ALLOWABLE NOISE=,F6.1//16H O-NOISE AV.(1) ,
       1F8.1,7H   (2) ,F8.1/16H X-NOISE AV.(1) ,F8.1,7H   (2) ,F8.1)
       WRITE(6,54)ID,IR
54     FORMAT(//1X,I4,8H SAMPLES,5X,I5,12H REJ.(NOISE)//8H REJECTS,
       12X,6HHEIGHT,2X,6HAV. AO,2X,6HAV. AX,2X,5HAX/AO,4X,2HED)
       HT=58.5
C   WRITE OUT TABLE
       DO 53 I=1,21
       HT=HT+1.5
       WRITE(6,58)IRJ(I),HT,AVAO(I),AVAX(I),XO(I),EL(I)
58     FORMAT(3X,I4,3X,F5.1,3X,F6.1,2X,F6.1,2X,F5.2,3X,F6.0)
53     CONTINUE
       GO TO 10
       STOP
       END
```

```
CHHHHHHHHHHHHHHHHHHHHHHHHH---HEAD---HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
C       HEAD SETS UP AND PRINTS OUT ONE LINE OF INFORMATION AS A
C  HEADING FOR A NEW PAGE.
CHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
C
       SUBROUTINE HEAD(J)
       REAL REAS(12)
       IF(J) 6,6,10
10     WRITE(6,25) REAS
25     FORMAT(1H1/1X,12A5)
       RETURN
6      WRITE(6,100)
100    FORMAT(1H1/27H GIVE 1 LINE OF INFORMATION)
       READ(4,200) REAS
200    FORMAT(12A5)
       RETURN
       END
```

```
        .TITLE  READ DATA IN DUMP MODE
/ZZZZZZZZZZ ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
/ .READ, DUMP MODE FROM DECTAPE ON A VARIABLE .DAT SLOT
/FILLS 252 DEC WORD BUFFER AND OUTPUTS 26
/WORDS TO ARRAY IDAT EVERY TIME CALLED.
/THESE ARE UNPACKED FROM 18 WORDS OF THE BUFFER.
/ IDAT: WORD  1         I.D. #
/       WORD 2-6        NOISE SAMPLES
/       WORD 7-27       DATA
/ NEGF:         SET IF A NEGATIVE NUMBER WAS IN THE DATA
/
        .GLOBL  DINIT,DUMPT,.DA
DUMP=4                          /TYPE OF I/O MODE
DATIN=2                         /.DAT SLOT TO READ DATA FROM
NSAM=32                         /# OF SAMPLES PER SET
NSAMP=NSAM/2+1                  /SIZE OF ONE SET PACKED DOUBLE
DTBLK=374                       /SIZE OF ONE BLOCK OF STORAGE
DATSTR=DTBLK/NSAMP              /# OF SETS PER ONE BLOCK OF STORAGE
NDPC=NSAM/2                     /NUMBER OF STORED DATA PAIRS PER SET
        .IODEV  DATIN
DINIT   0
        .INIT   DATIN,0,DINIT   /INITIALIZE DEVICE STORING THE DATA
        LAW     -1              /PREPARE TO READ
        DAC     CTT9            /    IN ONE BLOCK OF DATA
        LAC     BUF31           /RESET THE BUFFER POINTER WITH
        DAC     POINT2          /    THE ADDR. OF BUF3
        LAC     (ISZ    SWITC   /PREPARE TO READ TWO
        DAC     LCA             /    DUMMY BLOCKS
        JMP*    DINIT           /END OF INITIALIZATION
DUMPT   0
        JMS*    .DA             /PICKUP ADDR OF ADDR
        JMP     .+3             /OF ARRAY
A2      0
FLAG    0                       /SET ON NEG #
        LAC*    A2              /GET ADDR.
        DAC     A2              /    OF ARRAY
        LAW     -NDPC           /SET COUNTER OF DATA TO BE
        DAC     COUNT           /    PROCESSED
        ISZ     CTT9            /GET POINTER
        JMP     LBB             /NO, CONTINUE WITH PRESENT SET OF DATA
        LAW     -DATSTR         /RESET COUNTER TO THE NUMBER
        DAC     CTT9            /    OF SETS PER BLOCK OF STORAGE
LBA     .READ   DATIN,DUMP,BUF3,DTBLK   /GET 1 BLOCK OF DATA
C14     .WAIT   DATIN
LCA     ISZ     SWITC           /INITIALLY READ TWO DUMMY BLOCKS
        JMP     LBA
        LAW     -3              /RESET CONTROL TO READ
        DAC     SWITC           /    TWO DUMMY BLOCKS
        LAC     (JMP    LCB     /READ ONE BLOCK OF DATA
        DAC     LCA             /    AT A TIME
LCB     LAC     BUF31           /GET ADDRESS OF BUF3
        DAC     POINT2          /    POINT2 TO BUF3
LBB     LAC*    POINT2          /GET THE ID (FIRST WORD IN DATA SET)
        SAD     SEVN            /END OF FILE ID?
        JMP     ENF             /YES, RESET PARAM'S AND CLOSE FILE
        LAC*    POINT2          /GET ID AND PUT
        DAC*    A2              /    INTO THE FORTRAN ARRAY
        ISZ     A2              /GO TO NEXT ADDR. OF THE ARRAY
LOOP    ISZ     POINT2          /GO TO NEXT DATA WORD
        LAW     -2              /PREPARE TO UNPAC
        DAC     TC10            /    TWO DATA WORDS
        LAC*    POINT2          /GET DATA WORDS FROM BUF3
        SWHA                    /FIRST WORD IN LEFT HALF
UNPLP   AND     (777            /SAVE ONE DATA WORD
        SNA                     /CHECK FOR NEG. NUMBER
        ISZ*    FLAG            /SET IF NEG. NUMBER FOUND
        DAC*    A2              /LOAD # INTO FORTRAN ARRAY
        LAC*    POINT2          /GET DATA WORD AGAIN
        ISZ     A2              /GO TO NEXT LOC. IN ARRAY
        ISZ     TC10            /UNPACED TWO WORDS?
        JMP     UNPLP           /NO, LOOP AROUND
        ISZ     COUNT           /YES, HAS 34 DATA WORDS BEEN UNPACKED?
        JMP     LOOP            /NO, REPEAT UNPACKING PROCESS
OUT2    ISZ     POINT2          /YES, GO TO NEXT ID
        JMP*    DUMPT           /RETURN
SWITC   LAW     -3
COUNT   LAW     -NDPC
CTT9    LAW     -DATSTR
SEVN    376002
TC10    0
POINT2  .DSA    BUF3
BUF31   .DSA    BUF3
BUF3    .BLOCK  DTBLK
/-------------------------------------------------------
```

```
/
/ END OF FILE ROUTINE
ENF     LAC     SEVN                    /SET LAST ID TO
        DAC*    A2                      /     130050 DECIMAL
C15     .CLOSE  DATIN                   /CLOSE FILE
RT2     JMP*    DUMPT                   /RETURN TO PROC. PROGRAM
/
/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
        .END
```

```
C***************************** CALC ******************************
C       FROM GIVEN COLLISION FREQUENCIES,CALC ALONG WITH ELDEN
C CALCULATES THE CONSTANT VALUES USED IN THE ELECTRON DENSITY
C EQUATION GIVEN BY PIRNAT IN AERONOMY REPORT 29 AND WRITES THE
C PROGRAM CALC2 WHICH CALCULATES THE ELECTRO DENSITIES FOR THE
C PARTIAL-REFLECTION PROGRAMS.
C******************************************************************
        DIMENSION ARRAY(21),P(21),R(3),CF(3),EL(20),CALC2(2)
        DATA CALC2(1),CALC2(2)/5HCALC2,4H SRC/
        IDAT=1
        LABL=0
C       COLLISION FREQUENCY PROFILES
C       COLLISION FREQUENCY PROFILE FOR THE SUMMER
100     P(1)=192.3
        P(2)=156.9
        P(3)=127.5
        P(4)=102.7
        P(5)=82.37
        P(6)=66.25
        P(7)=52.50
        P(8)=41.66
        P(9)=32.81
        P(10)=25.84
        P(11)=20.1
        P(12)=15.53
        P(13)=11.89
        P(14)=9.057
        P(15)=6.817
        P(16)=5.399
        P(17)=3.827
        P(18)=2.862
        P(19)=2.124
        P(20)=1.563
        P(21)=1.180
C   WRITE THE PROGRAM HEADING ONTO TAPE
        CALL ENTER(IDAT,CALC2)
        WRITE(IDAT,10)
10      FORMAT(69H CCCCCCCCCCCCCCCCCCCCCCCCCCC---CALC2---CCCCCCCCCC
        1CCCCCCCCCCCCCCCCCCC/59H C      CALC2 IS A LIST OF CONSTANTS
        2 CALUATED FROM THE PROGRAMS/63H C CALC AND ELDEN AND CONT
        3AINS THE FUNCTION THAT CALCULATES THE/69H C ELECTRON DENSITIES
        4 FOR THE PARTIAL-REFLECTION PROCESSING PROGRAMS./40H C THE PRO
        5GRAM CALC WRITES THIS PROGRAM./69H CCCCCCCCCCCCCCCCCCCCCCCCCC
        6CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC/37H  SUBROUTINE
        7 CALC2(ARRAY,LL,LH,FD,IA)/39H  DIMENSION ARRAY(21),RATIO2
        8(21),FD(21)/57H C   GET THE PREDETERMINED CONSTANTS FOR THE
        9 RIGHT SEASON/19H      IF(IA)200,300,100/
        128H C  CONSTANTS FOR THE SUMMER)
        GO TO 400
C       COLLISION FREQUENCY PROFILE FOR THE WINTER
200     P(1)=133.5
        P(2)=107.8
        P(3)=87.12
        P(4)=70.04
        P(5)=56.33
        P(6)=45.28
        P(7)=36.55
        P(8)=29.32
        P(9)=23.52
        P(10)=18.80
        P(11)=14.97
        P(12)=11.99
        P(13)=9.561
        P(14)=7.541
        P(15)=6.008
        P(16)=4.748
        P(17)=3.758
        P(18)=2.941
        P(19)=2.321
```

```
      P(20)=1.810
      P(21)=1.431
      WRITE(IDAT,11)
11    FORMAT(11H      GO TO 400/
      128H C  CONSTANTS FOR THE WINTER)
      GO TO 400
C     COLLISION FREQUENCY PROFILE FOR THE EQUINOX
300   P(1)=160.2
      P(2)=130.2
      P(3)=105.3
      P(4)=84.90
      P(5)=68.25
      P(6)=54.75
      P(7)=43.57
      P(8)=34.31
      P(9)=27.07
      P(10)=21.32
      P(11)=16.82
      P(12)=13.26
      P(13)=10.33
      P(14)=8.062
      P(15)=6.246
      P(16)=4.835
      P(17)=3.758
      P(18)=2.915
      P(19)=2.260
      P(20)=1.733
      P(21)=1.359
      WRITE(IDAT,12)
12    FORMAT(11H      GO TO 400/
      125H C  CONSTANTS FOR EQUINOX)
C   SET THE COLLISION FREQUENCIES TO THE RIGHT ORDER OF MAGNITUDE
400   DO 401 I=1,21
      P(I)=P(I)*(10.**5)
401   CONTINUE
C   STATEMENT LABLE FOR THE NEW PROGRAM
      LABL=LABL+100
      J1=J1+4
      K=0

      DO 20 I=1,20
      K=K+1
      CF(1)=P(K)
      CF(2)=P(K+1)
C   CALCULATE CONSTANTS FOR THE ELECTRON DENSITY EQUATION
20    CALL ELDEN(R,CF,ARRAY(I),ARRAY(I+1),EL(I))
      ARRAY(1)=ARRAY(2)/ARRAY(1)
C   WRITE FIRST CONTSTANT WITH A STATEMENT LABLE
      WRITE(IDAT,405)LABL,ARRAY(1)
405   FORMAT(1H ,I3,12H      RATIO2( 1)=,F10.7)
C   WRITE THE REST OF THE RO AND RX CONSTANTS
      DO 25 I=2,20
      ARRAY(I)=ARRAY(I+1)/ARRAY(I)
25    WRITE(IDAT,410) I,ARRAY(I)
410   FORMAT(9H      RATIO2(,I2,2H)=,F10.7)
C   WRITE THE CONSTANT DENOMINATORS
      DO 30 I=1,20
30    WRITE(IDAT,420) I,EL(I)
420   FORMAT(5H      FD(,I2,2H)=,E13.6)
C   CALCULATE THE REST OF THECONSTANTS
      IF(J1.LT.5)GO TO 200
      IF(J1.LT.10)GO TO 300
C   WRITE THE REST OF THE PROGRAM TO CALCULATE ELECTRON DENSITIES
      WRITE(IDAT,40)
40    FORMAT(18H 400  DO 10 I=LL,LH/48H      IF(ARRAY(I).EQ.0..OR.
      1ARRAY(I+1).EQ.0.)GO TO 50/59H C   THE FUNCTION FOR THE CALCULA
      2TION OF ELECTRON DENSITIES/
      351H    FD(I)=ALOG((ARRAY(I)/ARRAY(I+1))*RATIO2(I))/FD(I)/
      410H    GO TO 10/12H 50 FD(I)=0./12H 10 CONTINUE/8H     RETURN)
      CALL CLOSE(IDAT)
      STOP
      END


C::::::::::::::::::::::-SUBROUTINE ELDEN-:::::::::::::::::::::::::::::::
C       DURING DATA PROCESSING THERE ARE ONLY 2 VARIABLES
C FOR EACH HEIGHT (AO AND AX).  THE EQUATION FOR THE
C ELECTRON DENSITY AS GIVEN BY BIRLY (1971) IS:
C ED=LN(((AX(1)/AO(1))/(RX(1)/RO(1)))/((AX(2)/AO(2))/(RX(2)/RO(2))))/FD
C WHERE LN IS THE NATURAL LOG AND 1 AND 2 ARE HEIGHT 1 AND 2
C       SUBROUTINE ELDEN CALCULATES THE CONSTANTS RX,RO,AND FD
C FOR EACH HEIGHT.
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
        SUBROUTINE ELDEN(AXBYAO,GNU,RXR01,RXR02,FD)
        DIMENSION AXBYAO(3),RXBYRO(3),RX(3),RO(3),GNU(3),RATIO(3)
C       APPROX INTEGRAL PARAMETERS
        A4=2.3983474E-2
        A3=1.1287513E+1
        A2=1.1394160E+2
        A1=2.4653115E+1
        B6=1.8064128E-2
        B5=9.3877372
        B4=1.4921254E+2
        B3=2.8958085E+2
        B2=1.2049512E+2
        B1=2.4656819E+1
        D3=1.1630641
        D2=1.6901002E+1
        D1=6.5945939
        E5=4.3605732
        E4=6.4093464E+1
        E3=6.8920505E+1
        E2=3.5355257E+1
        E1=6.6314497
        AXBYAO(3)=0
C       GNU(3) IS MEAN COLLISION FREQUENCY AT THE INTERMEDIATE HEIGHT
C       CALCULATE C INTEGRALS AT BOTH HEIGHTS AND FOR AVERAGE GNU
        DUM=GNU(1)+GNU(2)
        GNU(3)=0.5*DUM
        DO 22 K=1,3
        O=(2.59614E+7)/GNU(K)
        X=7.3886E+6/GNU(K)
        CTN=O*(O*(O*(O+A1)+A2)+A3)+A4
        CTD=O*(O*(O*(O*(O*(O+B1)+B2)+B3)+B4)+B5)+B6
        CTO=CTN/CTD
        CTXN=X*(X*(X*(X+A1)+A2)+A3)+A4
        CTXD=X*(X*(X*(X*(X*(X+B1)+B2)+B3)+B4)+B5)+B6
        CTX=CTXN/CTXD
        CFO=(O*(O*(O+D1)+D2)+D3)/(O*(O*(O*(O*(O+E1)+E2)+E3)+E4)+E5)
        CFX=(X*(X*(X+D1)+D2)+D3)/(X*(X*(X*(X*(X+E1)+E2)+E3)+E4)+E5)
C       CALCULATE RATIOS
        RX(K)=SQRT((X*CTX)**2+(2.5*CFX)**2)
        RO(K)=SQRT((O*CTO)**2+(2.5*CFO)**2)
        RXBYRO(K)=RX(K)/RO(K)
        RATIO(K)=AXBYAO(K)/RXBYRO(K)
22      CONTINUE
C       CALCULATE FD FROM FINAL VALUES OF DO LOOP
        FO=(5.*3.1824E+3*CFO)/(4.*3.0E+8*GNU(3))
        FX=(5.*3.1824E+3*CFX)/(4.*3.0E+8*GNU(3))
        FD=(FX-FO)*3.0E+9
        RXR01=RXBYRO(1)

        RXR02=RXBYRO(2)
        RETURN
        END
```

```
C***************************-CHECK-**********************************
C       PROGRAM READS IN THE NUMBER OF SAMPLES ASKS FOR BY OPERATOR.
C IF THE NUMBER IS ZERO, 31 NUMBERS ARE READ IN AND SET UP AS PARTIAL
C REFLECTION DATA IS (I.E. 5 NOISE SAMPLES AND 21 DATA POINTS & 5 EXTRA)
C DATA IS PRINTED OUT IN THE FORM OF ONE NUMBER PER HEIGHT AFTER EACH
C GROUP OF 26 SAMPLES ARE READ IN.  THIS HAS BEEN USE TO CHECK THE
C RECIEVER AND A/D CONVERTER AGAINST THE REYNOLDS SYSTEM AND TO SEE
C IF EVERYTHING IS OPERATING AS IT SHOULD.  IF THE NUMBER READ IN IS
C NOT ZERO, THAT NUMBER OF SAMPLES ARE READ FROM THE A/D CONVERTER
C AND AN AVERAGE OF ALL THE NUMBERS ARE TAKEN AND PRINTED OUT.
C********************************************************************
C
        DIMENSION IA(50),RA1(50),RA2(50)
        MAX=50
        I31=31
        WRITE (6,110)
110     FORMAT(10H ADC CHECK)
C   DEFAULT VALUE FOR THE # OF SAMPLES = 31
4       NS=I31
C   READ # OF SAMPLE TO BE READ FROM A/D CONVERTER
5       READ (4,210) IDV
210     FORMAT(I5)
        IF(IDV.NE.0)NS=IDV
        IF(NS.NE.I31)GO TO 50
```

```
C   FOR 31 NUMBERS READ IN, THE FORM USED IS 2 SETS OF 31 SAMPLES
C   AS IN THE PARTIAL REFLECTION COLLECTION
      I1=0
25    ICH=0
      I1=I1+1
      CALL INPAD(IA,NS,ICH)
6     IF(ICH.EQ.0) GO TO 6
      IF(I1.GT.1)GO TO 11
      DO 13 I=1,I31
C   CONVERSION ALGORTHOM FOR A/D CONVERTER NEG. #'S TO COMPUTER
C   NEGATIVE NUMBERS
      IF(IA(I).GT.511)IA(I)=3072+(4096+32768)*7+IA(I)
13    RA1(I)=FLOAT(IA(I))/.511
      GO TO 25
11    WRITE(6,101)
C   DO LOOP FOR SECOND SET OF NUMBERS READ IN
      DO 15 I=1,I31
      IF(IA(I).GT.511)IA(I)=3072+(4096+32768)*7+IA(I)
      HT=45.+FLOAT(I-1)*1.5
      IF(I.EQ.6.OR.I.EQ.11)WRITE(6,105)
      IF(I.EQ.11)WRITE(6,106)
      RA2(I)=FLOAT(IA(I))/.511
C   WRITE OUT THE NUMBERS IN AN ORDERLY WAY
15    WRITE(6,100) HT,RA1(I),RA2(I)
100   FORMAT(3X,F4.1,4HKM  ,F5.0,4HMV  ,F5.0,2HMV)
101   FORMAT(6H NOISE)
105   FORMAT(25H ------------------------)
106   FORMAT(5H DATA)
      GO TO 5
C   THE FOLLOWING DUMPS THE AVERAGE OF THE A/D CONVERTER NUMBERS
C   AND ALSO GIVES THE VALUE IN MILLIVOLTS
50    INS=(NS+MAX-1)/MAX
      TNS=INS*MAX

      DO 60 J=1,INS
      ICH=0
      CALL INPAD(IA,MAX,ICH)
20    IF(ICH.EQ.0) GO TO 20
      DO 55 I=1,MAX
      IF(IA(I).GT.511)IA(I)=3072+(4096+32768)*7+IA(I)
55    AV=AV+FLOAT(IA(I))
60    CONTINUE
      AV=AV/TNS
      AVV=AV/.512
      WRITE6,120)AV,AVV
120   FORMAT(9H AVERAGE=,F7.3,12H    VOLTAGE=,F8.3,3H MV)
      GO TO 5
      STOP
      END
```

```
        .TITLE  A/D CONVERTER SERVICE ROUTINES FOR BG.-FG.
/       RFKM15 V3A SERVICE ROUTINES FOR THE HP 5610A A TO D
/  CONVERTER.  THESE ROUTINES PERMIT INPUT OF ANY SPECIFIED
/  NUMBER OF SAMPLES INTO A CORE BUFFER.  INPUT MAY BE OVER-
/  LAPPED WITH PROGRAM EXCUTION, AND CONTROL MAY BE RELINQUISHED
/  TO LOWER PRIORITY PROGRAMS WHILE DATA TRANSFER TAKES PLACE.
/    MACRO-15 CALLING SEQUENCE:
/       JMS     INPAD
/       NUMBER OF SAMPLES REQUIRED
/       BUFFER ADDRESS
/       COMPLETION FLAG ADDRESS
/       REAL-TIME SUBROUTINE ADDRESS, PRIORITY LEVEL IN BITS 0-2
/   (EXAMPLE: 500000+RTSUBA)
/   (RETURNS HERE IMMEDIATELY)
/       IF THE 4TH WORD AFTER THE JMS IS 0, NO REAL-TIME SUBROUTINE
/   WILL BE ACTIVATED.  NOTE:  THE PRIORITY CODE FOR MAINSTREAM IS 1
/       THE COMPLETION FLAG IS CLEARED BY THE CALL TO INPAD,
/   AND SET TO +1 FOR NORMAL COMPLETION OR -1001 IF A DATA
/   TIMING ERROR OCCURS.
/
/
ADWCR=26                        /A-D WORD COUNT
ADCAR=ADWCR+1                   /AND CURRENT ADDRESS REGISTERS
.SCOM=100                       /MONITOR'S COMMUNICATION AREA
ADWI=703724                     /A-D CONVERTER WRITE INITIALIZE
ADSO=703701                     /SKIP ON WORD COUNT OVERFLOW
ADST=703721                     /SKIP ON DATA TIMING ERROR
ADCO=703704                     /CLEAR OVERFLOW FLAG
ADCT=703744                     /CLEAR TIMING FLAG
```

```
/
/  ENTRY POINT FOR A-D INTERFACE INITIALIZATION
/
        .GLOBL  INPAD,.DA
INPAD   0
        JMS*    .DA
        JMP     .+4
INAR    0
INWC    0
INFLAG  0
INR     JMP     INSET              /REPLACED BY "LAC*     INWC"
        TCA
        DAC*    (ADWCR)            /SET WORD COUNT
        LAW     -1
        TAD*    INAR               /BUFFER ADDRESS -1
        DAC*    (ADCAR)            /   TO CURRENT ADDRESS REG.
        DZM*    INFLAG             /CLEAR FLAG
        DZM     INSUB#             /CLEAR REAL-TIME SUBROUTINE
        ADWI                       /INITIALIZE INTERFACE
        JMP*    INPAD              /RETURN
/
/   THE FOLLOWING CODE IS EXECUTED ONLY ONCE
INSET   LAC*    (.SCOM+55)         /GET ENTRY POINT ADDERSS OF .SETUP
ADSVA   DAC     .
        JMS*    .-1                /CALL .SETUP TO CONNECT ADINT TO API
        ADSO
        ADINT
        DZM*    (204

        LAC     (LAC*   INWC
        DAC     INR                /MODIFY INSTRUCTION
        JMP     INR                / AND JUMP TO IT
/
/
/INTERRUPT SERVICE ROUTINE.  EXECUTED IMMEDIATELY AFTER COMPLETION
/  OF DATA TRANSFER.  DETERMINES STATUS OF A-D INTERFACE, SETS
/  COMPLETION FLAG AND ACTIVATES REAL-TIME SUBROUTINE.
/  RUNS AT API LEVEL 0.
/
/
ADINT   0
        DBA                        /PAGE ADDRESSING MODE
        DAC     ADSVA              /SAVE AC
        ADST                       /TIMING ERROR?
        SKP!CLA!IAC                /NO,+1 TO AC
        LAW     -1001              /YES, ERROR CODE
        DAC*    INFLAG             /SET FLAG
        ADCO                       /CLEAR
        ADCT                       /   INTERFACE FLAGS
ADXIT   LAC     ADSVA              /RESTORE AC
        DBR                        /SET TO LEAVE HARDWARE API LEVEL
        JMP*    ADINT              /RETURN TO INTERRUPTED PROGRAM
        .END
```