The Marshall Engineering Thermosphere (MET) Model
Volume I: Technical Description

R.E. Smith
Physitron, Inc., Huntsville, Alabama

EXECUTIVE SUMMARY

Volume I of this report presents a technical description of the Marshall Engineering Thermosphere (MET) model atmosphere and a summary of its historical development. Various programs developed to augment the original capability of the model are discussed in detail. The report also describes each of the individual subroutines developed to enhance the model. Computer codes for these subroutines are contained in four appendices.

Volume II contains a copy of each of the reference documents.

If you have questions or comments or need to request additional information, contact the Chief, Electromagnetics and Aerospace Environments Branch, Systems Analysis and Integration Laboratory, MSFC, AL 35812.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

<u>Title</u> <u>Page</u>

TABLE OF ILLUSTRATIONS

<u>Figure</u> <u>Title</u> <u>Page</u>

I. INTRODUCTION

The Marshall Engineering Thermosphere (MET) Model of the Earth's atmosphere at orbital altitudes was developed over a long period of time. This document summarizes the primary activities that occurred during the development and maturation of the model and the method and time various model options should be applied in space vehicle development programs.

II. SCOPE

The first volume of this document includes a description of the Earth's neutral orbital atmosphere, summary of the historical development of the model, and descriptions of various options that can be exercised with the model. Detailed descriptions of these options, when and how they can be used, and program listings are included in the appendices. Reference documents are contained in Volume II.

III. DESCRIPTION OF THE NEUTRAL THERMOSPHERE

The region of the Earth's atmosphere between about 90 and 500 kilometers altitude is known as the thermosphere. The region above about 500 kilometers is known as the exosphere. The temperature in the lower thermosphere increases rapidly with increasing altitude from a minimum at 90 kilometers, but eventually becomes altitude-independent at upper thermospheric altitudes. This asymptotic temperature, known as the exospheric temperature, does not vary with height at any given time due to the extremely short thermal conduction time. However, the exospheric temperature does vary with time because of solar activity and other factors discussed below.

State of the neutral thermosphere is most conveniently described in terms of a mean, with spatial and temporal variations about that mean. The neutral thermosphere is important for two reasons. First, even at its low density, it produces torques and drag on orbiting spacecraft. Second, the density-height profile of the atmosphere above 100 kilometers altitude modulates the flux of trapped radiation encountered at orbital altitudes.

A. VARIATIONS

1. Solar Activity

Short wavelength solar electromagnetic radiation (EUV and UV) changes substantially with level of solar activity. Thus, thermospheric density is

strongly dependent on the level of solar activity. An average 11-year solar cycle variation exists as well as a 27-day variation in density related to the average 27-day solar rotation period. Variations, however, tend to be slightly longer than 27 days early in the solar cycle when active regions occur more frequently at higher solar latitudes and slightly shorter than 27 days later in the solar cycle when the active regions occur more frequently closer to the Sun's equator. Coronal holes and active longitudes also affect this average 27-day variation. Changes in the thermospheric density related to changes in level of solar (and geomagnetic) activity, e.g., flares, eruptions, coronal mass ejections (CMEs), and coronal holes (CHs), can begin almost instantaneously(minutes to hours), although more often a lag of a day or more occurs.

2. Geomagnetic Activity

Interaction of solar wind with the Earth's magnetosphere (referred to as geomagnetic activity) leads to a high latitude heat and momentum source for the thermospheric gases. Some of this heat and momentum is convected to low latitudes. Geomagnetic activity varies, usually has one peak in activity just prior to and another just after the peak activity of the solar cycle as defined by the 10.7-cm solar radio noise flux. Also, larger solar cycle peaks are associated with more intense geomagnetic activity. A seasonal variation of geomagnetic activity occurs with maxima in March (± 1 month) and September (± 1 month) each year. This variation is possibly related to the tilt of the Sun's rotational axis toward the Earth.

3. Diurnal

Rotation of the Earth induces a diurnal (24-hour period) variation (diurnal tide) in thermospheric temperature and density. Due to a lag in response of the thermosphere to the EUV heat source, density maximizes around 2 p.m. local solar time at orbital altitudes at a latitude approximately equal to the subsolar point. The lag decreases with decreasing altitude. Similarly, minimum density occurs between 3 and 4 a.m. local solar time at about the same latitude in the opposite hemisphere. In lowest regions of the thermosphere (120 km and below), where characteristic thermal conduction time is on the order of a day or more, the diurnal variation is not a predominant effect.

Harmonics of the diurnal tide are also induced in the Earth's atmosphere. A semi-diurnal tide (period of 12 hours) and a ter-diurnal tide (period of 8 hours) are important in the lower thermosphere (below about 160 km for the semi-diurnal tide and much lower for the ter-diurnal tide). Because of large damping effects of molecular viscosity, these diurnal harmonic tides are not important at orbital altitudes.

4. Semiannual

This variation is believed to be a conduction mode of oscillation driven by a semiannual variation in Joule heating in the high latitude thermosphere (as a consequence of a semiannual variation in geomagnetic activity). The variation is latitudinally independent and is modified by compositional effects. Amplitude of the variation is height dependent and variable from year to year with a primary minimum in July, primary maximum in October, secondary minimum in January, followed by a secondary maximum in April. Magnitude and altitude dependence of the semiannual oscillation vary considerably from one solar cycle to another. This variation is important at orbital altitudes.

5. Seasonal-latitudinal in Lower Thermosphere

These seasonal-latitudinal variations are driven in the thermosphere by the dynamics of the lower atmosphere (mesosphere and below). Amplitude of the variation maximizes in the lower thermosphere between about 105 and 120 kilometers and diminishes to zero around 200 kilometers. Although the temperature oscillation amplitude is quite large, corresponding density oscillation amplitude is small. This variation is not important at orbital altitudes.

6. Seasonal-latitudinal of Helium

Satellite mass spectrometers measured a strong increase in helium above the winter pole. Over a year the helium number density varies by a factor of 42 at 275 km, 12 at 400 km, and 3 or 4 above 500 km. Formation of the winter helium bulge is primarily due to effects of global scale winds that blow from the summer to the winter hemisphere. Amplitude of the bulge decreases with increasing levels of solar activity due to increased effectiveness of exospheric transport above 500 km carrying helium back to the summer hemisphere. Also, a very weak dependence exists of helium bulge amplitude on magnitude of the lower thermospheric eddy diffusivity.

7. Waves

Atmospheric waves have been detected in temperature and density measurements throughout the atmosphere from the ground to at least 510 km. These fluctuations, caused by gravity waves, are so named because they are primarily oscillations of the neutral gas for which the restoring force is gravity. A thermospheric gravity wave produces a corresponding wave in the ionosphere known as a traveling ionospheric disturbance.

Thermospheric gravity waves oscillate with periods typically of 30 minutes to several hours and have horizontal wavelengths from hundreds of kilometers to about 4000

km. Density amplitudes of the larger scale waves are greater at higher latitudes and smaller toward the equator. At approximately 200 km altitude typical values for density wave amplitudes are 15% at auroral latitudes and 5% at equatorial latitudes. Gravity wave amplitudes generally decrease at greater altitudes in the thermosphere due to dissipation by molecular processes. Larger scale waves survive to greater altitudes than the smaller.

8.   Winds

At low latitudes (less than 28.5°) wind speeds range from 100 to 200 meters per second. At high latitudes(greater than about 65°) speeds can be 1500 meters per second or more. Rapid (minutes) changes in wind direction (to 180°), probably driven by gravity waves, have been observed by a satellite.

B. SOLAR ACTIVITY PARAMETERS

Various surrogate indices are used to quantify levels of solar activity. One is the 10.7-cm solar radio noise flux, designated $F_{10.7}$. Although EUV radiation heats the atmosphere, this radiation cannot be measured on the ground. The $F_{10.7}$ can be measured on the ground and correlates well with the EUV radiation.

An index used as a measure of geomagnetic activity is the planetary geomagnetic activity index, a_p, (or k_p essentially the logarithm of a_p). This index is based on magnetic fluctuation data reported every 3 hours at 12 stations between geomagnetic latitudes 48° and 63° selected to provide good longitude coverage. Although high latitude ionospheric current fluctuations drive the magnetic field fluctuations observed at these stations, the magnetic field fluctuations do not drive the thermosphere. Thus, good correlations are not always found between observed density changes and the a_p index.

Figures 1 and 2 show the variability, during a period of low solar activity, of these two indices (a_p and $F_{10.7}$).

4
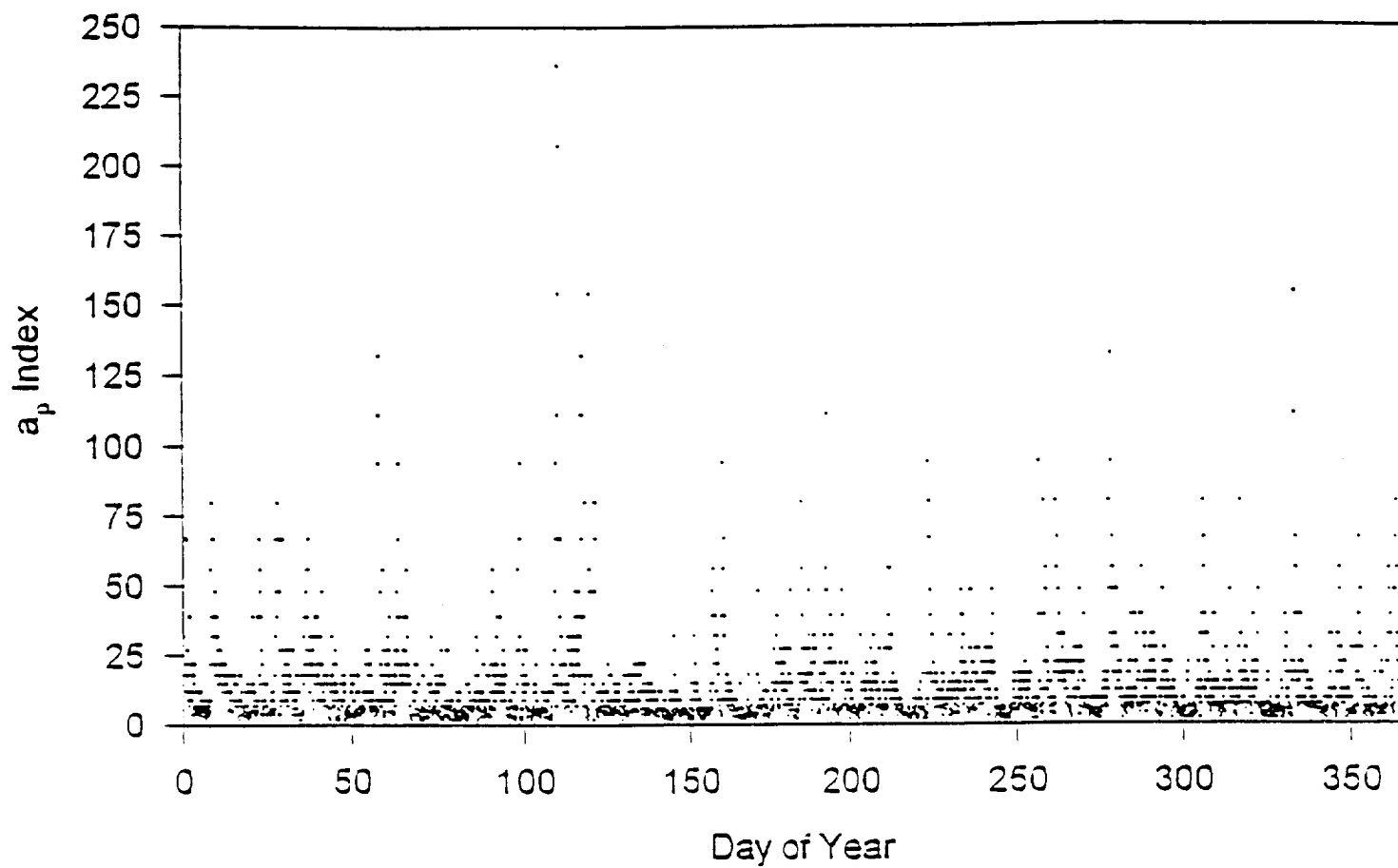
Figure 1. Variation in Geomagnetic Index, a_p, in 1985.
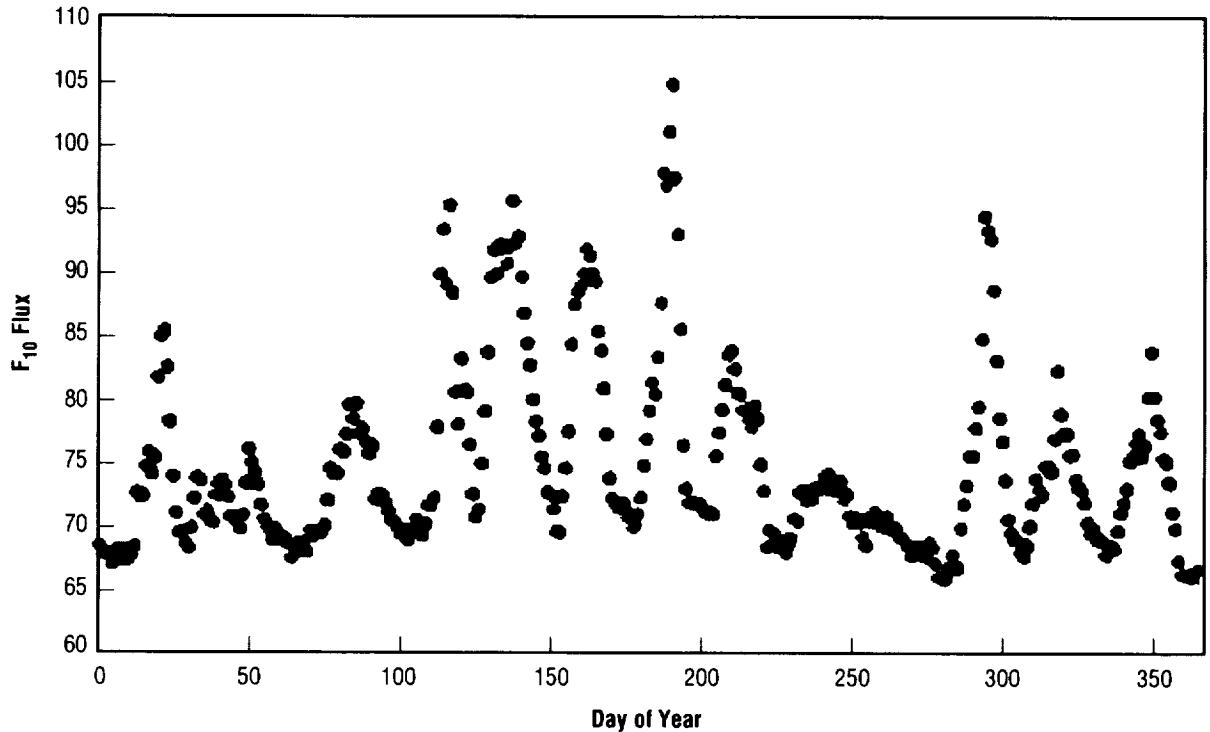
Figure 2. Variation in $F_{10.7}$ Flux in 1985.

IV. HISTORICAL DEVELOPMENT

A. JACCHIA 1964 MODEL

In the middle 1960s personnel at the NASA Marshall Space Flight Center (MSFC) responsible for predicting the orbital decay histories of satellites began studies to determine which atmospheric model when combined with the appropriate orbit propagation program would most accurately predict the observed decay histories of 39 satellites. Since the observed decay histories were already available, a number of available models of the thermosphere were selected for use in appropriate computer programs. Because the satellites had decayed prior to the start of the study, actual values of proxy input parameters required by the models were used. These proxy parameters were representative of actual solar conditions that occurred during the decay periods of the satellites. The Jacchia 1964 model had the best performance statistically and, therefore, was selected for use by MSFC.

B. JACCHIA 1970 MODEL

Almost coincidental with the completion of MSFC's study the Smithsonian Astrophysical Observatory (SAO) published the Jacchia 1970 model atmosphere (ref. 1).

Due to the great similarity between the 1964 and 1970 models and confirmation from observed data of the existence of major differences, the decision was made to use the newer model. In the 1964 model the temperature induced density bulge remained on the equator all year while the 1970 model bulge followed the latitudinal excursions of the Sun.

C. JACCHIA 1971 MODEL

In 1971 SAO published the Jacchia 1971 model (ref. 2). Although this model had several new features confirmed by observational data, overall, the model was not as representative of the atmosphere as the 1970 model.

D. MARSHALL ENGINEERING THERMOSPHERE (MET) MODEL

An independent study by environmental personnel at MSFC showed that the 1970 model could be improved by adding seasonal-latitudinal variations in the lower thermosphere density below 170 kilometers and adding seasonal-latitudinal variations in helium from the 1971 model above 500 kilometers altitude while retaining the orbital decay prediction accuracy of the 1964 model. Subsequent minor fairing modifications eliminated step function increases in density spuriously introduced by these modifications. This was the first version of the MET model used at MSFC. Additional minor modifications made in the late 1980s corrected programming errors in earlier computer programs and made the complete program more understandable and user friendly. For publications on programming changes and computer codes for the changed model see references 3, 4, and 5.

1. Statistical Analysis Mode, MET-SAM

This was the status quo for the MET model at the beginning of the Space Station Program when a meeting was held at the NASA Johnson Space Center (JSC) to establish design criteria for the development, testing, and operational phases of the Station. The complexities of the vehicle, requirement for a guaranteed 30-year orbital lifetime, and reboost strategy required to fulfill this requirement, plus the requirements of the various users, made this a most formidable task faced with very tight monetary restrictions. A complicated reboost strategy had to be developed to protect the astronauts from radiation at orbital altitudes and to ensure a 30-year lifetime without outside assistance. A limit existed on the altitude attainable by the Space Shuttle as well as the number of flights the Shuttle could make for on-orbit

construction of the Station and for resupply of fuel and reboost energy. This strategy affected the size of the propellant tanks on-board Space Station and Shuttle to transport additional fuel to the Station. Adequate precautions also had to be taken to ensure that the Space Station would not re-enter if the Shuttle missed a scheduled reboost event.

A critical factor in these deliberations at JSC was not only the representativeness of the atmospheric model but also the values of the input parameters which control model output. The atmospheric model required the 162-day mean values of 10.7-cm solar radio noise flux, previous day value of the 10.7-cm solar radio noise flux, and 3-hourly value of the geomagnetic index, K_p or a_p, 6.7 hours prior to time of analysis application. For any future time application the only available values of these three parameters were predictions of the 13-month smoothed values of the 10.7-cm solar radio noise flux and the geomagnetic index, a_p, based on statistics of the data during all previous solar cycles. Although statistics of separate distributions of each of the three parameters existed, nothing was available on probabilities of combined occurrences of the three parameters. As a result of this void, the decision at the JSC meeting was to use different values of these two parameters, the 13-month smoothed values of 10.7-cm solar radio noise flux and the a_p during development, testing, and operation phases of the Space Station program.

Early in the Station developmental phase, the need for a detailed investigation of the combined occurrence of these three parameters became evident. A study was begun to determine exactly how these three proxy input parameters should be combined for use in future time period activities. The result of this study is the Statistical Analysis Mode, MET-SAM, described in reference 6.

2. Small Scale Thermosphere Simulations

Studies of the Guidance, Navigation & Control (GN&C) Systems for Space Station revealed some of the more recently discovered short time period phenomena in the thermosphere (not included in the MET model) were critical to the GN&C systems performance. The existence of waves in the thermosphere was discovered in the OGO 6 satellite density measurements. When the MET model was used in GN&C studies involving a space vehicle having widely separated centers of gravity and pressure, the effects of these waves were very significant. This significance was further enhanced by operational constraints placed on the Space Station by users—notably the micro-gravity researchers' requirements for 30 consecutive days of gravity levels below 10^{-6}gs. Since short period fluctuations could be the result of fluctuations in solar activity, the probability of occurrence of 30-day quiescent periods was a concern. Environmental specialists decided to add to MET a wave model. An in-depth analysis was made of observed density measurements to construct this numerical model of the wave spectra expected to occur during all levels of solar activity. Since design activities progressed rapidly and answers were needed for on-

going engineering analyses before the observational data were completely analyzed, representative sets were selected from the 10.7-cm solar activity and a_p databases for five different levels of solar activity that might exist during orbital operations of the Space Station. Data sets of both parameters were selected and tailored to reflect the 95 percentile profile that might exist during selected 90-day periods (length of time scheduled between Space Shuttle reboost and refueling missions). Results of this analysis of the interim procedure to be used in the engineering analyses until the in-depth wave study is completed are contained in reference 7.

3. Waves

Available observational data from the AE-E satellite in a low inclination $(19.7°)$ orbit were obtained from the National Space Sciences Data Center, GSFC, while programs were being written to accomplish the required analysis. The wave model described in reference 8 is a callable subroutine addition to the MET model.

4. Waves2

Almost at completion of this wave study and subsequent inclusion in the MET model, the announcement was made that the Space Station would be a joint effort between the USA and Russia and in a $51.6°$ inclination orbit. All previous analyses indicated most waves present in the thermosphere originated near the auroral ovals, propagated equatorward with phase fronts aligned with constant latitudes and with amplitudes that decreased as they propagated. Analyzed observational data from satellites in higher inclination orbits indicated a new wave model should be developed for inclusion in the MET model and applied to the International Space Station. The analysis and resulting callable subroutine for the MET model are described in reference 9.

V.   BASIC MET MODEL

The basic MET model, ref. 3, is an empirical static diffusion model with coefficients obtained from satellite drag analyses. With the proper input parameters, specified below, the exospheric temperature can be calculated. In the original development phase of the model the prime objective was to model the total neutral mass density of the thermosphere by adjusting temperature profiles until agreement between modeled and measured total densities was achieved. Agreement between modeled temperatures and temperatures measured on later missions was not always achieved. Thomson-scatter radar temperature measurements generally show that the diurnal temperature maximum lags the density maximum by a couple of hours, whereas in the MET model the temperature and the density maxima and minimal are in phase.

With exospheric temperature specified, the temperature can be calculated for any altitude between 90 and 2500 km from an empirically determined temperature profile. The density for all points on the globe at 90-km altitude is assumed constant and mixing of atmospheric constituents prevails to 105 km. Between these two altitudes the mean molecular mass varies as a result of dissociation of molecular oxygen to atomic. At 120-km altitude the ratio of atomic oxygen to molecular is assumed to be 1.5. Density between 90 and 105 km is calculated by integration of the barometric equation. For altitudes above 105 kilometers the diffusion equation for each of the individual species (O_2, O, N_2, He, and Ar) is integrated upward from the 105-km level. For hydrogen (H) integration of the diffusion equation proceeds upward from 500-km altitude. Total mass density is calculated by summing the individual specie mass densities.

The total mass density is modified further by the effects of seasonal-latitudinal density variation of the lower thermosphere below 170-km altitude and seasonal-latitudinal variations of helium (He) above 500km. These two effects were incorporated into the MET model using the equations developed by Jacchia for his 1971 thermospheric model (ref. 2).

The basic MET model includes all variations listed in Section III, with exception of the wind and wave variations. A single computer program exists for all applications of the basic MET model; however, two applications, listed below, require different values of the input parameters, also listed below. Appendix A is a listing of the FORTRAN source code for this basic MET model. A more detailed description of the model is in references 3, 4, and 5.

A. INPUT PARAMETERS FOR REAL-TIME OR AFTER-THE-FACT APPLICATIONS

1. Time: Year, month, day, hour, minute
2. Position: Altitude, geographic latitude and longitude
3. Solar Activity Parameters: Previous day's value of the 10.7-cm solar radio noise flux, centered value of the 10.7-cm solar radio noise flux averaged over 6 solar rotations (162 days), a_p or K_p geomagnetic index 6 to 7 hours prior to time of application.

B. INPUT PARAMETERS FOR FUTURE TIME APPLICATIONS

1. Time: Year, month, day, hour, minute
2. Position: Altitude, geographic latitude and longitude
3. Solar Activity Parameters   At present the only available data are predicted 13-month smoothed values of both the 10.7-cm solar radio noise flux and geomagnetic index, a_p.

C. OUTPUT PARAMETERS FOR ALL APPLICATIONS

1. Exospheric temperature
2. Temperature
3. N_2 number density
4. O_2 number density
5. O number density
6. Ar number density
7. He number density
8. H number density
9. Average molecular weight
10. Total mass density
11. Log_{10} mass density
12. Total pressure
13. Local gravity acceleration
14. Ratio specific heats
15. Pressure scale height
16. Specific heat constant p
17. Specific heat constant v

The total mass density, temperature, and individual species number densities all have the same phase variation in the MET model.

VI.   BASIC MET MODEL WITH INTERNAL GRAVITY WAVES

The shortest time period variations in the basic MET model occur every three hours as a result of the a_p variations, giving a Nyquist period of six hours. Tides with periods of twelve and eight hours are known to exist at thermospheric altitudes and have small amplitudes. The smallest time period for all of these variations far exceeds the average orbital period for near-Earth orbiting vehicles. The only known variations at thermospheric altitudes with periods less than an orbital period are those associated with internal gravity waves.

Atmospheric gravity waves exist from the troposphere to the top of the thermosphere. Their spatial and temporal scales are commensurate with their sources and both scales increase with increasing altitude. These waves can transport energy and momentum over large distances. Thermospheric gravity waves generally have spatial and temporal scales greater than 100 km and 20 minutes, respectively; however, waves with periods of several hours and wavelengths of several thousand kilometers have been inferred from observations.

11

Those waves, generated in situ at high latitudes in the thermosphere in response to geomagnetic activity, have larger scales and phase speeds. The phase speeds, as they propagate predominantly toward the equator, depend on the level of geomagnetic activity.

A. OPTIONAL WAVE MODELS

Optional additions to the basic MET model realistically model the observed wave spectra. Results of analyses of orbital altitude density measurements show total mass density in the thermosphere is best represented by summing the basic MET model total mass density and a quasi-random wave component having an absolute amplitude proportional to the mean basic MET model density.

1.   FOR VEHICLES WITH ORBITAL INCLINATIONS LESS THAN OR EQUAL TO 40° The FORTRAN source code is listed in Appendix B. See reference 8 for additional details.

   a. Input Parameters

      1. All input parameters for the basic MET model

      2. 13-month smoothed values of the 10.7-cm solar radio noise flux and geomagnetic index, A_p

   b. Output Parameters

      1. Amplitude of the wave expressed as a percentage of and to be added to the basic MET model total mass density

2.   FOR VEHICLES WITH ORBITAL INCLINATIONS GREATER THAN 40° Source code is listed in Appendix C. See reference 9 for additional details.

   a. Input Parameters

      1. All input parameters for the basic MET model

      2. A daily A_p value

   b. Output parameters

      1. Amplitude of the wave expressed as a percentage of and to be added to the basic MET model total mass density

VII. STATISTICAL ANALYSIS MODE, MET-SAM

The Statistical Analysis Mode (SAM) of the MET model was developed for use primarily during the design, development, and testing phases of a space vehicle when only statistical estimates of the 13-month smoothed mean values of the three solar parameters are required as inputs. While the exospheric temperature in MET-SAM is obtained in a different manner than in the basic MET model, densities for the same exospheric temperature are identical in both.

The MET-SAM provides answers to frequently asked questions during design and development of a space vehicle--what is the percentage of time that the recommended density value will be exceeded during the operational phase of the vehicle? Or, how confident are you that the recommended density value will not be exceeded more than a specified percentage of time? The answers are crucial in the design of the guidance and control capability, selection of the altitude at which the vehicles will orbit, amount of shielding necessary to protect the crew and vehicle from solar radiation and orbital debris, and reboost strategy for payloads put in low-Earth orbit by the Shuttle. MET-SAM is based on the premise that most applications during the development phase require detailed knowledge about maximum and minimum densities that will be encountered with limits on the magnitude of variations that occur during monthly time periods. With a slight modification MET-SAM has the capability to include the diurnal (daily) variation provided the input solar parameters remain constant during the orbit.

Initially a new database was constructed for 1947 through 1991. Contents were the eight 3-hourly values of a_p, daily value of 10.7-cm solar radio noise flux, and 162-day mean value of 10.7-cm solar radio noise flux centered on day of application referenced to the 13-month smoothed mean value of 10.7-cm solar radio noise flux centered on month of application. The global minimum, mean, and maximum exospheric temperatures were calculated for every three-hour period from 1947 through 1991 using algorithms in the MET model and appropriate solar activity input parameters from this new database. These temperatures were sorted into five levels of solar activity, as defined by the 13-month smoothed value of the 10.7-cm solar radio noise flux. Cumulative percentage frequency (CPF) distributions were calculated for these temperatures and levels of solar activity. Results of these calculations are given in Tables 1 through 18, Appendix D. Range of temperature in each distribution is the result of various combinations of three solar activity parameters that are required input parameters for the basic MET model. All uncertainties in how the three parameters should be combined are included in the distributions in the tables based on how they occurred in the historical records.

This procedure allows the design engineer to select the risk level he is willing to accept by selecting the appropriate temperature and density from the CPF distribu-

tions. A complete description is in reference 6 and computer program listing in Appendix D.

To use the MET-SAM:

A. INPUT PARAMETERS

1. Altitude, in kilometers

2. Appropriate temperature from CPF distributions tables

B. OUTPUT PARAMETERS

1. Exospheric temperature
2. Temperature at altitude
3. N_2 number density
4. O_2 number density
5. O number density
6. Ar number density
7. He number density
8. H number density
9. Average molecular weight
10. Total mass density
11. Log_{10} mass density
12. Total pressure
13. Local gravity acceleration
14. Ratio specific heats
15. Pressure scale height
16. Specific heat constant p
17. Specific heat constant v

VIII. REFERENCES

The following documents are contained in Volume II, References, which qualified requestors can obtain from Chief, Electromagnetics and Aerospace Environments Branch, System Analysis and Integration Laboratory, MSFC, AL 35812.

1.   Jacchia, L. G., New Static Models of the Thermosphere and Exosphere with Empirical Temperature Profiles. Smithsonian Astrophysical Observatory Special Report No. 313, 1970.

2.   Jacchia, L. G., Revised Static Models of the Thermosphere and Exosphere with Empirical Temperature Profiles. Smithsonian Astrophysical Observatory Special Report No. 332, 1971.

3.   Johnson, D. L. and R. E. Smith, The MSFC/J70 Orbital Atmosphere Model and the Data Bases for the MSFC Solar Activity Prediction Technique. NASA TM-86522, 1985.

4.   Hickey, M. P., The NASA Marshall Engineering Thermosphere Model. NASA CR-179359, 1988.

5.   Hickey, M. P., An Improvement in the Integration Procedure Used in the NASA Marshall Engineering Thermosphere Model. NASA CR-179389, 1988.

6.   Smith, R. E. and M. P. Hickey, The Marshall Engineering Thermosphere Model Atmosphere, Statistical Analysis Mode (MET-SAM). Physitron, Inc. Report PHY-92R032, 1992.

7.   Hickey, M. P. and R. E. Smith, Ninety-day Solar and Geomagnetic Activity Input Files for Thermospheric Variation Simulation: Simulation Data Files/Release 2. Physitron, Inc. Report PHY-92R031, 1992.

8.   Hickey, M. P., A Simulation of Small-Scale Thermospheric Density Variations for Engineering Applications. NASA CR-4605, 1994.

9.   Hickey, M. P., An Engineering Model for the Simulation of Small-Scale Thermospheric Density Variations for Orbital Inclinations Greater Than 40°. NASA CR 201140, 1996.

APPENDIX A

PROGRAM LISTING

MARSHALL ENGINEERING THERMOSPHERE (MET) MODEL

```
C*************************************************************************
**
C*                                          *
C*          The  Marshall  Space  Flight  Center          *
C*          Marshall Engineering  Thermosphere Model          *
C*      -------------------------------------------------      *
C*                                          *
C*                  written by              *
C*                                          *
C*              Mike Hickey              *
C*          Universities Space Research Association          *
C*              NASA / MSFC , ED44          *
C*              Tel. (205) 544-5692          *
C*                                          *
C*   This program is a driving program for the following subroutines :-   *
C*                                          *
C*              ATMOSPHERES              *
C*              SOLSET              *
C*              TIME              *
C*              J70          *
C*                                          *
C*   The atmospheric model is a modified Jacchia 1970 model and is given in   *
C*   the subroutine J70. All of the other subroutines were designed to      *
C*   allow flexible use of this model so that various input parameters could   *
C*   be varied within a driving program with very little software development.*
C*   Thus, for example, driving routines can be written quite easily to      *
C*   facilitate the plotting of output as line or contour plots. Control is   *
C*   achieved by setting the values of four switches in the driving program,   *
C*   as described in subroutine ATMOSPHERES.              *
C*                                          *
C*************************************************************************
**

      REAL*4  INDATA (12) , OUTDATA (12) , AUXDATA (5)

      CHARACTER*1  SWITCH (4)

C     CALL LIB$INIT_TIMER

C Set all switches to 'Y' so that only one particular calculation is performed

          SWITCH (1) = 'Y'
          SWITCH (2) = 'Y'
          SWITCH (3) = 'Y'
          SWITCH (4) = 'Y'

      CALL ATMOSPHERES ( INDATA, OUTDATA, AUXDATA, SWITCH )
```

C Now type output data

```
      WRITE(1,*)' All output in MKS units'
      WRITE(1,*)' '
      WRITE(1,*)' Exospheric temperature = ', OUTDATA (1),' K'
      WRITE(1,*)' Temperature          = ', OUTDATA (2),' K'
      WRITE(1,*)' N2 number density    = ', OUTDATA (3),' /m3'
      WRITE(1,*)' O2 number density    = ', OUTDATA (4),' /m3'
      WRITE(1,*)' O number density     = ', OUTDATA (5),' /m3'
      WRITE(1,*)' A number density     = ', OUTDATA (6),' /m3'
      WRITE(1,*)' He number density    = ', OUTDATA (7),' /m3'
      WRITE(1,*)' H number density     = ', OUTDATA (8),' /m3'
      WRITE(1,*)' Average molecular wt. = ', OUTDATA (9)
      WRITE(1,*)' Total mass density   = ', OUTDATA (10),' kg/m3'
      WRITE(1,*)' Log10 mass density   = ', OUTDATA (11)
      WRITE(1,*)' Total pressure       = ', OUTDATA (12),' Pa'
      WRITE(1,*)' Local grav. acceln.  = ', AUXDATA (1),' m.sec-2'
      WRITE(1,*)' Ratio specific heats = ', AUXDATA (2)
      WRITE(1,*)' Pressure scale-height = ', AUXDATA (3),' m'
   WRITE(1,*)' Specific heat cons. p = ', AUXDATA (4),'m2.sec-2.K-1'
   WRITE(1,*)' Specific heat cons. v = ', AUXDATA (5),'m2.sec-2.K-1'
      WRITE(1,*)' '

C     CALL LIB$SHOW_TIMER

      STOP
      END
```

SUBROUTINE ATMOSPHERES (INDATA, OUTDATA, AUXDATA, SWITCH)

```
C*****************************************************************************
**
C*                     DESCRIPTION:-                     *
C*                     -----------                    *
C*                                                  *
C* Calculate atmospheric data in single precision using subroutine J70    *
C* and J70SUP.                                       *
C*                                              *
C*                     SUBROUTINES:-                     *
C*                     -----------                   *
C*                                                *
C*     TIME, SOLSET, GMC, J70 and J70SUP                     *
C*                                              *
C*                         INPUT:-                    *
```

```
C*              ------                          *
C* ------------ all single precision, either through  ---------------- *
C* ------------ subroutines or from main driver prog.  ---------------- *
C*                                              *
C* INDATA (1) -- altitude        = Z                    *
C*  .. (2) -- latitude       = XLAT                           *
C*  .. (3) -- longitude      = XLNG                           *
C*  .. (4) -- year (yy)       = IYR                         *
C*  .. (5) -- month (mm)      = MN                            *
C*  .. (6) -- day (dd)        = IDA                         *
C*  .. (7) -- hour (hh)       = IHR                         *
C*  .. (8) -- mins (mm)        = MIN                          *
C*  .. (9) -- geomagnetic index    = IGEO_IND                 *
C*  .. (10)-- solar radio noise flux= F10                     *
C*  .. (11)-- 162-day average F10   = F10B                    *
C*  .. (12)-- geomagnetic activity index = GI=AP              *
C*                                                            *
C*                                                            *
C*                   OUTPUT:-                     *
C*                   -------                    *
C*                                          *
C*     NOTE :  All output in MKS units           *
C*     -------------------------------       *
C*                                          *
C* -------------------- all single precision  ---------------------- *
C*                                                      *
C*                                          *
C* OUTDATA (1) -- exospheric temperature (K)             *
C*  .. (2) -- temperature at altitude Z               *
C*  .. (3) -- N2 number density  (per meter-cubed)           *
C*  .. (4) -- O2 number density (    ..    )        *
C*  .. (5) -- O  number density (    ..   )        *
C*  .. (6) -- A  number density (    ..   )        *
C*  .. (7) -- He number density (    ..   )      *
C*  .. (8) -- H  number density (    ..   )        *
C*  .. (9) -- average molecular weight             *
C*  .. (10)-- total density                    *
C*  .. (11)-- log10 ( total density )         *
C*  .. (12)-- total pressure ( Pa )         *
C*                                              *
C* AUXDATA (1) -- gravitational acceleration ( m/s-s )      *
C*  .. (2) -- ratio of specific heats           *
C*  .. (3) -- pressure scale-height ( m )       *
C*  .. (4) -- specific heat at constant pressure      *
C*  .. (5) -- specific heat at constant volume       *
C*                                          *
C*                                          *
C*                   COMMENTS:-                          *
C*                   ---------                       *
```

```
C*                                        *
C*  SWITCH(1) -- if Y(es), date and time are input from terminal through     *
C*                      subroutine TIME once only                         *
C*  SWITCH(2) -- if Y(es), solar/magnetic activity are input from terminal    *
C*                      through subroutine SOLSET once only                   *
C*  SWITCH(3) -- if Y(es), only ONE altitude value is input from terminal    *
C*                      through main calling program                      *
C*  SWITCH(4) -- if Y(es), only ONE latitude AND longitude are input from    *
C*                      terminal through main calling program          *
C*                                                                  *
C*   ATMOSPHERES written by Mike Hickey ( USRA, NASA/ED44 )              *
C*              Tel: (205) 544-5692                    *
C*      -------- January-April 1987 ---------                *
C**************************************************************************************
**


        EXTERNAL TIME

        DIMENSION  AUXDATA (5)

        INTEGER  HR

        REAL*4  LAT, LON, INDATA (12), OUTDATA (12)

        CHARACTER*1  SWITCH (4)

        PARAMETER PI = 3.14159265


C --------------------------------------------------------------------------------
C       This next section is only executed on the first call to ATMOSPHERES

        DO WHILE ( CALL. EQ. 0.0 )

C SECTION A:-
C ----------

        IF ( SWITCH(1). EQ. 'Y' ) THEN

        CALL TIME ( IYR, MON, IDA, HR, MIN, SWITCH(1) )
        INDATA (4) = FLOATJ (IYR)
        INDATA (5) = FLOATJ (MON)
        INDATA (6) = FLOATJ (IDA)
        INDATA (7) = FLOATJ (HR)
        INDATA (8) = FLOATJ (MIN)

        END IF

C SECTION B:-
```

```
C ----------

        IF ( SWITCH(2). EQ. 'Y' ) THEN

        CALL SOLSET ( IGEO_IND, F10, F10B, GI, SWITCH(2) )
        INDATA (9) = FLOATJ (IGEO_IND)
        INDATA (10) = F10
        INDATA (11) = F10B
        INDATA (12) = GI

        END IF

C  SECTION C:-
C ----------

        IF ( SWITCH(3). EQ. 'Y' ) THEN

        TYPE *,' Input altitude, km'
        ACCEPT *, INDATA (1)
        Z = INDATA (1)

        END IF

C  SECTION D:-
C ----------

        IF ( SWITCH(4). EQ. 'Y' ) THEN

        TYPE *,' Input latitude and longitude, degrees'
        ACCEPT *, ( INDATA(I), I= 2,3 )
        LAT = INDATA (2)
        LON = INDATA (3)
        RLT = INDATA (2) * PI / 180.  !  geographic latitude, radians

        END IF


        CALL = 1.0
                END DO

C       End of first executable section
C ----------------------------------------------------------------------------

C  The following depend on the values of the switches

C****
C* SECTION 1:-

                IF ( SWITCH(1). NE. 'Y' ) THEN
```

```
            IYR = JINT ( INDATA (4) )
            MON = JINT ( INDATA (5) )
            IDA = JINT ( INDATA (6) )
            HR  = JINT ( INDATA (7) )
            MIN = JINT ( INDATA (8) )
            CALL TIME ( IYR, MON, IDA, HR, MIN, SWITCH(1) )

                 END IF


C*****
C* SECTION 2:-

                 IF ( SWITCH(2). NE. 'Y' ) THEN

            IGEO_IND = JINT ( INDATA (9) )
            F10     = INDATA (10)
            F10B    = INDATA (11)
            GI      = INDATA (12)
            CALL SOLSET ( IGEO_IND, F10, F10B, GI, SWITCH(2) )

                 END IF

C*****
C* SECTION 3:-

                 IF ( SWITCH(3). NE. 'Y' ) THEN

            Z = INDATA (1)

                 END IF


C*****
C* SECTION 4:-

                 IF ( SWITCH(4). NE. 'Y' ) THEN

            LAT = INDATA (2)
            LON = INDATA (3)
            RLT = INDATA (2) * PI / 180.  ! geographic latitude, radians

                 END IF


C  All setting-up complete.
```

```
          CALL J70 ( INDATA, OUTDATA )
          CALL J70SUP ( Z, OUTDATA, AUXDATA )


          RETURN

          ENTRY ATMOS_ENT ( DUMMY )
          CALL = DUMMY
          RETURN

          END
```

```
          SUBROUTINE TIME ( IYR, MON, IDA, HR, MIN, SWITCH )
```

```
C*********************************************************************
**
C*                                          *
C*        This subroutine sets up time of year and day          *
C*                                          *
C*              INPUTS/OUTPUTS:                    *
C*              ----------------              *
C* IYR  =  year ( 2 digits )                 *
C* MON  =  month                             *
C* IDA  =  day of month                      *
C* HR   =  hour of day                       *
C* MIN  =  minutes                           *
C*                                  *
C*        Written by Mike Hickey, USRA              *
C*********************************************************************
**
```

```
          DIMENSION  IDAY ( 12 )

          INTEGER  HR

          CHARACTER*1  SWITCH



          PARAMETER PI = 3.14159265
          DATA IDAY / 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 /
C-------------------------------------------------------------------
C ------   If  SWITCH = Y(es)  then input data and time from terminal -------
```

C_____

—

```
        IF ( SWITCH.EQ.'Y'. OR. SWITCH.EQ.'y' )  THEN

TYPE *, ' Input date and time of date? ( yy,mm,dd,hh,mm ) '
ACCEPT *, IYR, MON, IDA, HR, MIN

        END IF
```

C_____

```
IF ( JMOD (IYR,4) .EQ. 0 )  THEN
        IF ( JMOD (IYR,100) .NE. 0 ) IDAY ( 2 ) = 29
ELSE
        IDAY ( 2 ) = 28
END IF

DAYTOT = 0.0

        DO 1 I = 1, 12
DAYTOT = DAYTOT + FLOATJ ( IDAY ( I ) )
1       CONTINUE

        IF ( MON. GT. 1 )              THEN

KE = MON - 1
ID = 0
        DO 2 I = 1, KE
ID = ID + IDAY (I)
2       CONTINUE

ID = ID + IDA
                                       ELSE

DD = IDA
        END IF


RETURN
END
```

.

```
SUBROUTINE SOLSET ( IGEO_IND, F10, F10B, GI, SWITCH )
```

A9

```
C*********************************************************************
**
C*                                              *
C* This subroutine simply calls for a setup of the solar-activty and auroral  *
C* activity indices.                                  *
C*                                              *
C*              INPUTS/OUTPUTS:                        *
C*              ----------------                  *
C* IGEO_IND = geomagnetic index                        *
C* F10     = solar radio noise flux                    *
C* F10B    = 162-day average F10                        *
C* GI      = geomagnetic activity index                *
C*                                          *
C*     Written by Mike Hickey, USRA                    *
C*********************************************************************
**

      CHARACTER*1  SWITCH

      IGEO_IND = 2


C ---------------------------------------------------------------
C ----- If SWITCH = Y(es) then input geomagnetic indices from terminal -------
C_____
--

      IF ( SWITCH.EQ.'Y'. OR. SWITCH.EQ.'y' ) THEN

C       TYPE *, ' Input geomagnetic index ( 1-KP, 2-AP ) '
C       ACCEPT *, IGEO_IND

        TYPE *, ' Input solar radio noise flux ( F10 = 0-400 ) '
        ACCEPT *, F10

        TYPE *, ' Input 162-day average F10 ( F10B = 0-250 ) '
        ACCEPT *, F10B

C            IF ( IGEO_IND . EQ. 2 )            THEN
C
C       TYPE *, ' Input geomagnetic activity index ( GI = 0-400 ) '
C
C                                   ELSE
C
C       TYPE *, ' Input geomagnetic activity index ( GI = 0-9 ) '
C
C            END IF

        TYPE *,' Input AP index ( AP = 0 - 400 ) '
```

```
        ACCEPT *, GI

        END IF

C
```
_____

```
        RETURN
        END
```

SUBROUTINE J70SUP (Z, OUTDATA, AUXDATA)

```
C****************************************************************************
**
C*                                    *
C*              DESCRIPTION:-                  *
C*              -----------                  *
C*                                    *
C* J70SUP calculates auxilliary variables which are output in array    *
C* AUXDATA, given data input from J70 which are contained in array OUTDATA.  *
C*                                    *
C*              INPUT DATA:-                   *
C*              ----------                  *
C*                                    *
C* Z -- altitude (km)                     *
C* TZZ -- temperature at altitude z  = OUTDATA (2)          *
C*     -- N2 number density     =   .. (3)          *
C*     -- O2 ..    ..    =   .. (4)          *
C*     -- O  ..    ..    =   .. (5)          *
C*     -- A  ..    ..    =   .. (6)          *
C*     -- He ..    ..    =   .. (7)          *
C*     -- H  ..    ..    =   .. (8)          *
C* EM -- average molecular weight  =   .. (9)          *
C* DENS -- total density      =   .. (10)         *
C* P -- total pressure      =   .. (12)         *
C*                                    *
C*              OUTPUT DATA:-                  *
C*              -----------                  *
C*                                    *
C* G -- gravitational acceleration    = AUXDATA (1)        *
C* GAM -- ratio of specific heats     = AUXDATA (2)        *
C* H -- pressure scale-height      = AUXDATA (3)        *
C* CP -- specific heat at constant pressure = AUXDATA (4)          *
```

```
C*  CV -- specific heat at constant volume   = AUXDATA (5)          *
C*                                              *
C*  Written by Mike Hickey, USRA                         *
C*********************************************************************
**
```

```
        REAL*4  OUTDATA (12), AUXDATA (5), H

        G = 9.80665 / ( ( 1. + Z / 6.356766E3 )**2 )

        H = OUTDATA (12) / ( G * OUTDATA (10) )

        SUM1 = OUTDATA (3) + OUTDATA (4)
        SUM2 = 0.0
            DO 1 I = 5, 8
        SUM2 = SUM2 + OUTDATA (I)
1           CONTINUE

        GAM = ( 1.4 * SUM1 + 1.67 * SUM2 ) / ( SUM1 + SUM2 )

        CV = G * H / (  ( GAM - 1.0 ) * OUTDATA (2)  )

        CP = GAM * CV

        AUXDATA (1) = G
        AUXDATA (2) = GAM
        AUXDATA (3) = H
        AUXDATA (4) = CP
        AUXDATA (5) = CV

        RETURN
        END
```

```
        SUBROUTINE J70 ( INDATA, OUTDATA )

C*********************************************************************
**
C**                                       **
C**              J70 developed from J70MM by            **
C**                  Mike P. Hickey                **
C**          Universities Space Research Association        **
C**                    at                  **
C**          NASA / Marshall Space Flight Center, ED44,        **
C**              Huntsville, Alabama, 35812, USA.         **
C**                 Tel. (205) 544-5692             **
```

```
C**                                                    **
C** INPUTS:      through the subroutine calling list        **
C**                                                    **
C** OUTPUTS:     through the subroutine calling list          **
C**                                                    **
C**                                                    **
C**               INPUT DATA:                     **
C**               ------------                       **
C**     Z   -- altitude          = INDATA (1)          **
C**     XLAT -- latitude         = INDATA (2)           **
C**     XLNG -- longitude        = INDATA (3)           **
C**     IYR -- year (yy)         = INDATA (4)          **
C**     MN -- month (mm)         = INDATA (5)           **
C**     IDA -- day  (dd)         = INDATA (6)          **
C**     IHR -- hour (hh)         = INDATA (7)          **
C**     MIN -- mins (mm)         = INDATA (8)           **
C**     I1 -- geomagnetic index  = INDATA (9)           **
C**     F10 --  solar radio noise flux  = INDATA (10)        **
C**     F10B --  162-day average F10    = INDATA (11)          **
C**     GI -- geomagnetic activity index = INDATA (12)          **
C**                                                    **
C**                                                    **
C**               OUTPUT DATA:                      **
C**               -------------                      **
C**     T-- exospheric temperature     = OUTDATA (1)         **
C**     TZZ-- temperature at altitude Z = OUTDATA (2)          **
C**     A(1)-- N2 number density       = OUTDATA (3)         **
C**     A(2)-- O2 number density       = OUTDATA (4)         **
C**     A(3)-- O  number density       = OUTDATA (5)         **
C**     A(4)-- A  number density       = OUTDATA (6)         **
C**     A(5)-- He number density       = OUTDATA (7)         **
C**     A(6)-- H  number density       = OUTDATA (8)         **
C**     EM-- average molecular weight  = OUTDATA (9)           **
C**     DENS-- total density           = OUTDATA (10)        **
C**     DL-- log10 ( total density )   = OUTDATA (11)        **
C**     P-- total pressure             = OUTDATA (12)         **
C**                                                    **
C** NB. Input through array 'INDATA'                 **
C**     Output through array 'OUTDATA'                **
C*******************************************************************************
**


      DIMENSION  A ( 6 )

          REAL*4  INDATA ( 12 ), OUTDATA ( 12 )
          PARAMETER RGAS = 8.31432E3             ! J/kmol-K
          PARAMETER BFH = 440.0

C Calcultions performed for only one latitude , one longitude
```

```
C and one altitude

C
C** Set parameters to INDATA values
C
        Z   = INDATA (1)
        XLAT = INDATA (2)
        XLNG = INDATA (3)
        IYR  = JINT ( INDATA (4) ) + 1900
        MN   = JINT ( INDATA (5) )
        IDA  = JINT ( INDATA (6) )
        IHR  = JINT ( INDATA (7) )
        MIN  = JINT ( INDATA (8) )
        I1   = JINT ( INDATA (9) )
        F10  = INDATA (10)
        F10B = INDATA (11)
        GI   = INDATA (12)


        CALL TME ( MN , IDA , IYR , IHR , MIN , XLAT , XLNG , SDA ,
     .           SHA , DD , DY )

        CALL TINF ( F10 , F10B , GI , XLAT , SDA , SHA , DY , I1 , TE )

     CALL JAC ( Z , TE , TZ , A(1) , A(2) , A(3) , A(4) , A(5) , A(6) ,
     .           EM , DENS , DL )


        DENLG = 0.
        DUMMY = DL
        DEN   = DL

        IF ( Z .LE. 170. ) THEN
                CALL SLV ( DUMMY , Z , XLAT , DD )
                DENLG = DUMMY
        END IF

C
C** 'Fair' helium number density between base fairing height ( BFH ) and 500 km
C

        IF ( Z. GE. 500. ) THEN
                  CALL SLVH ( DEN , A(5) , XLAT , SDA )
                  DL = DEN
        ELSE IF ( Z .GT. BFH ) THEN
                  DHEL1 = A ( 5 )
                  DHEL2 = A ( 5 )
                  DLG1 = DL
                  DLG2 = DL
```

```
            CALL SLVH ( DLG2 , DHEL2 , XLAT , SDA )
                  IH = Z
      CALL FAIR5 ( DHEL1 , DHEL2 , DLG1 , DLG2 , IH , FDHEL , FDLG )
                  DL = FDLG
                  A ( 5 ) = FDHEL
      END IF

            DL = DL + DENLG
            DENS = 10.**DL
            XLAT = XLAT * 57.29577951


C  Fill OUTDATA array
            OUTDATA (1) = TE
            OUTDATA (2) = TZ

            DO 80 I = 1, 6
            OUTDATA (I+2) = 1.E6 * ( 10. ** A(I) )
80                CONTINUE

            OUTDATA (9) = EM
            OUTDATA (10) = DENS * 1000.
            OUTDATA (11) = DL
            P = OUTDATA (10) * RGAS * TZ / EM
            OUTDATA (12) = P

            RETURN
            END




      SUBROUTINE TME ( MN , IDA , IYR , IHR , MIN , XLAT , XLNG ,
                  SDA , SHA , DD , DY )


C***************************************************************************
**
C** Subroutine 'TME' performs the calculations of the solar declination    **
C** angle and solar hour angle.                              **
C**                                          **
C** INPUTS:  MN  = month                               **
C**       IDA = day                              **
C**       IYR = year                             **
C**       IHR = hour                             **
C**       MIN = minute                           **
C**       XMJD= mean Julian date                      **
C**       XLAT= latitude ( input-geocentric latitude )       **
```

A15

```
C**        XLNG= longitude ( input-geocentric longitude, -180,+180 )      **
C**                                                 **
C** OUTPUTS: SDA = solar declination angle (rad)                **
C**        SHA = solar hour angle (rad)                   **
C**        DD  = day number from 1 JAN.                   **
C**        DY  = DD / tropical year                   **
C**      Modified by Mike Hickey, USRA                   **
C**********************************************************************
**

       DIMENSION IDAY(12)


       PARAMETER YEAR = 365.2422
       PARAMETER A1 = 99.6909833
       PARAMETER A2 = 36000.76892
       PARAMETER A3 = 0.00038708
       PARAMETER A4 = 0.250684477
       PARAMETER B1 = 0.0172028
       PARAMETER B2 = 0.0335
       PARAMETER B3 = 1.407
       PARAMETER PI = 3.14159265
       PARAMETER TPI = 6.28318531
       PARAMETER PI2 = 1.57079633
       PARAMETER PI32 = 4.71238898
       PARAMETER RAD_DEG = 0.017453293
     DATA IDAY / 31,28,31 ,30,31,30 ,31,31,30 ,31,30,31 /
       XLAT = XLAT / 57.29577951
       YR = IYR

       IF ( JMOD(IYR,4) .EQ. 0 ) THEN
         IF ( JMOD(IYR,100) .NE. 0 ) IDAY(2) = 29  ! Century not a leap year
       ELSE
         IDAY(2) = 28
       END IF
            ID = 0
       IF ( MN. GT. 1 ) THEN
         DO 20 I = 1 , MN-1
            ID = ID + IDAY(I)
20       CONTINUE
       END IF
            ID = ID + IDA
            DD = ID
            DY = DD/YEAR
C
C** Compute mean Julian date
C
       XMJD = 2415020. + 365. * ( YR - 1900. ) + DD
            + FLOATJ ( ( IYR - 1901 ) / 4 )
```

A16

```
C
C** Compute Greenwich mean time in minutes GMT
C
        XHR = IHR
        XMIN = MIN
        GMT = 60 * XHR + XMIN
        FMJD = XMJD - 2435839. + GMT / 1440.
C
C** Compute Greenwich mean position - GP ( in rad )
C
        XJ = ( XMJD - 2415020.5 ) / ( 36525.0 )
        GP = AMOD ( A1 + A2 * XJ + A3 * XJ * XJ + A4 * GMT , 360. )


C
C** Compute right ascension point - RAP ( in rad )
C
C** 1st convert geocentric longitude to deg longitude -  west neg , + east
C
        IF ( XLNG .GT. 180. ) XLNG = XLNG - 360.

        RAP = AMOD ( GP + XLNG , 360. )


C
C** Compute celestial longitude - XLS ( in rad ) - - zero to 2PI
C
        Y1 = B1 * FMJD
        Y2 = 0.017202 * ( FMJD - 3. )
        XLS = AMOD ( Y1 + B2 * SIN(Y2) - B3 , TPI )
C
C** Compute solar declination angle - SDA ( in rad )
C
        B4 = RAD_DEG * ( 23.4523 - 0.013 * XJ )
        SDA = ASIN ( SIN ( XLS ) * SIN ( B4 ) )
C
C** Compute right ascension of Sun - RAS ( in rad ) - - zero to 2PI
C
C** These next few lines do not appear in NASA CR-179359 or NASA CR-??????
C** They are added here to ensure that that argument of ASIN stays bounded
C** between -1 and +1 , which could otherwise be effected by roundoff error.
        ARG = TAN ( SDA ) / TAN ( B4 )
                IF ( ARG .GT. 1.0 ) ARG = 1.0
                IF ( ARG .LT. -1. ) ARG = -1.0
        RAS = ASIN ( ARG )
C
C** Put RAS in same quadrant as XLS
C
        RAS = ABS ( RAS )
        TEMP = ABS ( XLS )
```

```
        IF ( TEMP.LE.PI .AND. TEMP.GT.PI2 )  THEN
              RAS = PI - RAS
        ELSE IF ( TEMP.LE.PI32 .AND. TEMP.GT.PI )  THEN
              RAS = PI + RAS
        ELSE IF ( TEMP.GT.PI32 )  THEN
              RAS = TPI - RAS
        END IF
              IF ( XLS. LT. 0. ) RAS = -RAS


C
C** Compute solar hour angle - SHA ( in deg ) - -
C
        SHA = RAP * RAD_DEG - RAS
        IF ( SHA.GT.PI ) SHA = SHA - TPI
        IF ( SHA.LT.-PI ) SHA = SHA + TPI



        RETURN
        END
```

```
        SUBROUTINE TINF ( F10 , F10B , GI, XLAT, SDA , SHA , DY , I1 ,TE )

C********************************************************************************
**
C** Subroutine 'TINF' calculates the exospheric temperature according to   **
C** L. Jacchia SAO 313, 1970                          **
C**                                      **
C**  F10 = solar radio noise flux ( x E-22 Watts / m2 )          **
C**  F10B= 162-day average F10                      **
C**  GI  = geomagnetic activity index                  **
C**  LAT = geographic latitude at perigee ( in rad )          **
C**  SDA = solar declination angle ( in rad )            **
C**  SHA = solar hour angle                      **
C**  DY  = D / Y ( day number / tropical year ) ; 1          **
C**  I1  = geomagnetic equation index  ( 1--GI=KP , 2--GI=AP )      **
C**  RE  = diurnal factor  KP, F10B, AVG                **
C**                                  **
C**    CONSTANTS -- C = solar activity variation            **
C**          -- BETA , etc = diurnal variation          **
C**          -- D = geomagnetic variation          **
C**          -- E = semiannual variation            **
C**                                  **
C** Modified by Mike Hickey, USRA                  **
```

```
C******************************************************************
**

      PARAMETER PI = 3.14159265
          PARAMETER TPI = 6.28318531
      PARAMETER XM = 2.5
          PARAMETER XNN = 3.0
C
C**  Ci are solar activity variation variables
C
      PARAMETER C1 = 383.0
              PARAMETER C2 = 3.32
              PARAMETER C3 = 1.80
C
C**  Di are geomagnetic variation variables
C
          PARAMETER D1 = 28.0
          PARAMETER D2 = 0.03
          PARAMETER D3 = 1.0
          PARAMETER D4 = 100.0
          PARAMETER D5 = -0.08
C
C**  Ei are semiannual variation variables
C
      PARAMETER E1 = 2.41
              PARAMETER E2 = 0.349
              PARAMETER E3 = 0.206
              PARAMETER E4 = 6.2831853
      PARAMETER E5 = 3.9531708
      PARAMETER E6 = 12.5663706
      PARAMETER E7 = 4.3214352
    PARAMETER E8 = 0.1145
              PARAMETER E9 = 0.5
              PARAMETER E10 = 6.2831853
      PARAMETER E11 = 5.9742620
      PARAMETER E12 = 2.16

      PARAMETER BETA = -0.6457718
      PARAMETER GAMMA = 0.7504916
      PARAMETER P = 0.1047198
      PARAMETER RE = 0.31
C
C** solar activity variation
C
      TC = C1 + C2 * F10B + C3 * ( F10 - F10B )
C
C** diurnal variation
C
      ETA   = 0.5 * ABS ( XLAT - SDA )
      THETA = 0.5 * ABS ( XLAT + SDA )
```

A19

```
        TAU    = SHA + BETA + P * SIN ( SHA + GAMMA )

        IF ( TAU. GT. PI ) TAU = TAU - TPI
        IF ( TAU. LT.-PI ) TAU = TAU + TPI

        A1 = ( SIN ( THETA ) )**XM
        A2 = ( COS ( ETA ) )**XM
        A3 = ( COS ( TAU / 2. ) )**XNN
        B1 = 1.0 + RE * A1
        B2 = ( A2 - A1 ) / B1
        TV = B1 * ( 1. + RE * B2 * A3 )
        TL = TC * TV
C
C** geomagnetic variation
C
        IF ( I1.EQ.1 ) THEN
                TG = D1 * GI + D2 * EXP(GI)
        ELSE
                TG = D3 * GI + D4 * ( 1 - EXP ( D5 * GI ) )
        END IF
C
C** semiannual variation
C
        G3 = 0.5 * ( 1.0 + SIN ( E10 * DY + E11 ) )
        G3 = G3 ** E12
        TAU1 = DY + E8 * ( G3 - E9 )
        G1 = E2 + E3 * ( SIN ( E4 * TAU1 + E5 ) )
        G2 = SIN ( E6 * TAU1+ E7 )
        TS = E1 + F10B * G1 * G2
C
C** exospheric temperature
C
        TE = TL + TG + TS

        RETURN
        END




        SUBROUTINE JAC ( Z , T , TZ , AN , AO2 , AO , AA , AHE , AH , EM ,
                         DENS , DL )

C***********************************************************************
**
C**                                            **
C**  Subroutine 'JAC' calculates the temperature TZ , the total density DENS **
```

```
C**  and its logarithm DL, the mean molecular weight EM, the individual    **
C**  specie number densities for N, O2, O, A, HE and H ( each preceded with **
C**  an 'A' ) at altitude Z given the exospheric temperature T.         **
C**  This subroutine uses the subroutine 'GAUSS' and the function          **
C**  subprograms 'TEMP' and 'MOL_WT'.                            **
C**                                        **
C**   Rewritten by Mike Hickey, USRA                         **
C************************************************************************************
**


         DIMENSION  ALPHA(6) , EI(6) , DI(6) , DIT(6)
         REAL*4  MOL_WT

         PARAMETER AV = 6.02257E23
         PARAMETER QN  = .78110
         PARAMETER QO2 = .20955
         PARAMETER QA  = .009343
         PARAMETER QHE = 1.289E-05
         PARAMETER RGAS = 8.31432
         PARAMETER PI  = 3.14159265
         PARAMETER T0  = 183.


C        GRAVITY ( ALTITUDE ) = 9.80665/((1. + ALTITUDE/6.356766E3)**2)



         DATA ALPHA / 0.0 , 0.0 , 0.0 , 0.0 , -.380 , 0.0 /
     DATA EI / 28.0134 , 31.9988 , 15.9994 , 39.948 , 4.0026 ,1.00797 /



         TX = 444.3807 + .02385 * T - 392.8292 * EXP ( -.0021357 * T )
         A2 = 2. * (T-TX) / PI
         TX_T0 = TX - T0
         T1 =  1.9 * TX_T0 /  35.
         T3 = -1.7 * TX_T0 / ( 35.**3 )
         T4 = -0.8 * TX_T0 / ( 35.**4 )
         TZ = TEMP ( Z , TX , T1 , T3 , T4 , A2 )


C**      SECTION 1
C**      ---------

         A = 90.
         D = AMIN1 ( Z , 105. )

C Integrate gM/T from 90 to minimum of Z or 105 km :-

         CALL GAUSS ( A, D, 1, R, TX , T1 , T3 , T4 , A2 )
```

C The number 2.1926E-8 = density x temperature/mean molecular weight at 90 km.

```
EM = MOL_WT ( D )
TD = TEMP ( D , TX , T1 , T3 , T4 , A2 )

DENS = 2.1926E-8 * EM * EXP( -R / RGAS ) / TD

FACTOR = AV * DENS
PAR = FACTOR / EM
FACTOR = FACTOR / 28.96
```

C For altitudes below and at 105 km calculate the individual specie number
C densities from the mean molecular weight and total density.

```
IF ( Z. LE. 105 )  THEN

DL = ALOG10 ( DENS )
AN  = ALOG10 ( QN * FACTOR )
AA  = ALOG10 ( QA * FACTOR )
AHE = ALOG10 ( QHE * FACTOR )
AO  = ALOG10 ( 2. * PAR * ( 1.-EM / 28.96 ) )
AO2 = ALOG10 ( PAR * ( EM * ( 1.+QO2 ) / 28.96-1. ) )
AH  = 0.
```
C
C** Return to calling program
C
```
RETURN


END IF
```

C**   SECTION 2 : This section is only performed for altitudes above 105 km
C**   ---------

C Note that having reached this section means that D in section 1 is 105 km.
C ----

C Calculate individual specie number densities from the total density and mean
C molecular weight at 105 km altitude.

```
DI(1) = QN * FACTOR
DI(2) = PAR * (EM * (1.+QO2) / 28.96-1.)
DI(3) = 2. * PAR * (1.- EM / 28.96)
DI(4) = QA * FACTOR
DI(5) = QHE * FACTOR
```

C Integrate g/T from 105 km to Z km :-

```
        CALL GAUSS ( D, Z, 2, R, TX , T1 , T3 , T4 , A2 )

                DO  41  I = 1 , 5
        DIT(I) = DI(I) * ( TD / TZ ) **(1.+ALPHA(I)) * EXP( -EI(I) *
     &  R / RGAS)
                IF ( DIT(I). LE. 0. ) DIT(I) = 1.E-6
41              CONTINUE




C**  This section calculates atomic hydrogen densities above 500 km altitude.
C**  Below this altitude , H densities are set to 10**-6.

C**     SECTION 3
C**     ---------

        IF ( Z .GT. 500. ) THEN

                A1 = 500.
                S = TEMP ( A1 , TX , T1 , T3 , T4 , A2 )

        DI(6) = 10.** ( 73.13 - 39.4 * ALOG10 (S) + 5.5 * ALOG10(S)
     &      *ALOG10(S))

        CALL GAUSS ( A1, Z, 7, R, TX , T1 , T3 , T4 , A2 )

                DIT(6) = DI(6) * (S/TZ) * EXP ( -EI(6) * R / RGAS )

        ELSE

                DIT (6) = 1.0

        END IF




C  For altitudes greater than 105 km , calculate total density and mean
C  molecular weight from individual specie number densities.

                DENS=0
                DO  42  I = 1 , 6
                        DENS = DENS + EI(I) * DIT(I) / AV
42              CONTINUE

        EM = DENS * AV / ( DIT(1)+DIT(2)+DIT(3)+DIT(4)+DIT(5)+DIT(6) )
                DL = ALOG10 (DENS)

                AN  = ALOG10(DIT(1))
```

```
AO2 = ALOG10(DIT(2))
AO  = ALOG10(DIT(3))
AA  = ALOG10(DIT(4))
AHE = ALOG10(DIT(5))
AH  = ALOG10(DIT(6))



        RETURN
        END
```

```
        FUNCTION TEMP ( ALT , TX , T1 , T3 , T4 , A2 )
```

```
C*********************************************************************
**
C**                                          **
C** Function subprogram 'TEMP' calculates the temperature at altitude ALT  **
C** using equation (10) for altitudes between 90 and 125 km and equation   **
C** (13) for altitudes greater than 125 km , from SAO Report 313.      **
C**                                          **
C** Written by Mike Hickey, USRA                      **
C*********************************************************************
**
```

```
        PARAMETER BB = 4.5E-6

                U = ALT - 125.
        IF ( U .GT . 0. ) THEN
        TEMP = TX  +  A2 * ATAN ( T1 * U * ( 1. + BB * (U**2.5)) / A2 )
                ELSE
                TEMP = TX  +  T1 * U  +  T3 * (U**3)  +  T4 * (U**4)
        END IF

        END
```

```
        REAL*4 FUNCTION MOL_WT ( A )
```

```
C*********************************************************************
**
C**                                          **
C** Subroutine 'MOL_WT' calculates the molecular weight for altitudes   **
C** between 90 and 105 km according to equation (1) of SAO report 313.  **
```

```
C**  Otherwise, MOL_WT is set to unity.                    **
C**                                          **
C**  Written by Mike Hickey, USRA                   **
C************************************************************************
**

      DIMENSION  B(7)

      DATA B / 28.15204 , -0.085586, 1.284E-4, -1.0056E-5, -1.021E-5,
     .        1.5044E-6, 9.9826E-8 /

      IF ( A. GT. 105. ) THEN

          MOL_WT = 1.

      ELSE

          U = A - 100.
          MOL_WT = B (1)
      DO  1  I = 2 , 7

      MOL_WT = MOL_WT + B (I) * U ** ( I-1 )

1        CONTINUE

      END IF

      END




      SUBROUTINE GAUSS ( Z1 , Z2 , NMIN , R , TX , T1 , T3 , T4 , A2 )

C************************************************************************
**
C**  Subdivide total integration-altitude range into intervals suitable for  **
C**  applying Gaussian Quadrature , set the number of points for integration **
C**  for each sub-interval , and then perform Gaussian Quadrature.       **
C**     Written by Mike Hickey, USRA, NASA/MSFC, ED44, July 1988.        **
C************************************************************************
**

      REAL*4  ALTMIN (9) , C(8,6), X(8,6), MOL_WT
      INTEGER  NG (8) , NGAUSS , NMIN , J

      GRAVITY ( ALTITUDE ) = 9.80665 / ( ( 1. + ALTITUDE /
     &   6.356766E3 )**2 )
```

```
      DATA  ALTMIN / 90., 105., 125., 160., 200., 300., 500., 1500.,
 &    2500. /
      DATA  NG  / 4 , 5 , 6 , 6 , 6 , 6 , 6 , 6 /

C  Coefficients for Gaussian Quadrature ...

      DATA  C / .5555556 , .8888889 , .5555556 , .0000000 , ! n=3
   .          .0000000 , .0000000 , .0000000 , .0000000 , ! n=3
   .          .3478548 , .6521452 , .6521452 , .3478548 , ! n=4
   .          .0000000 , .0000000 , .0000000 , .0000000 , ! n=4
   .          .2369269 , .4786287 , .5688889 , .4786287 , ! n=5
   .          .2369269 , .0000000 , .0000000 , .0000000 , ! n=5
   .          .1713245 , .3607616 , .4679139 , .4679139 , ! n=6
   .          .3607616 , .1713245 , .0000000 , .0000000 , ! n=6
   .          .1294850 , .2797054 , .3818301 , .4179592 , ! n=7
   .          .3818301 , .2797054 , .1294850 , .0000000 , ! n=7
   .          .1012285 , .2223810 , .3137067 , .3626838 , ! n=8
   .          .3626838 , .3137067 , .2223810 , .1012285 / ! n=8

C  Abscissas for Gaussian Quadrature ...

      DATA  X / -.7745967 , .0000000 , .7745967 , .0000000 , ! n=3
   .          .0000000 , .0000000 , .0000000 , .0000000 , ! n=3
   .          -.8611363 , -.3399810 , .3399810 , .8611363 , ! n=4
   .          .0000000 , .0000000 , .0000000 , .0000000 , ! n=4
   .          -.9061798 , -.5384693 , .0000000 , .5384693 , ! n=5
   .          .9061798 , .0000000 , .0000000 , .0000000 , ! n=5
   .          -.9324695 , -.6612094 , -.2386192 , .2386192 , ! n=6
   .          .6612094 , .9324695 , .0000000 , .0000000 , ! n=6
   .          -.9491079 , -.7415312 , -.4058452 , .0000000 , ! n=7
   .          .4058452 , .7415312 , .9491079 , .0000000 , ! n=7
   .          -.9602899 , -.7966665 , -.5255324 , -.1834346 , ! n=8
   .          .1834346 , .5255324 , .7966665 , .9602899 / ! n=8


          R = 0.0

      DO 2  K = NMIN , 8

          NGAUSS = NG (K)
      A    = ALTMIN (K)
      D    = AMIN1 ( Z2 , ALTMIN (K+1) )
      RR   = 0.0
      DEL = 0.5 * ( D - A )
      J    = NGAUSS - 2

      DO 1  I = 1 , NGAUSS

          Z = DEL * ( X(I,J) + 1. ) + A
      RR = RR + C(I,J) * MOL_WT(Z) * GRAVITY(Z) / TEMP ( Z,TX,T1,T3,
```

```
     &   T4,A2 )

1        CONTINUE

         RR = DEL * RR
         R  = R + RR
         IF ( D .EQ. Z2 ) RETURN

2        CONTINUE

         RETURN
         END




         SUBROUTINE SLV ( DEN , ALT , XLAT , DAY )

C*********************************************************************
**
C** Subroutine 'SLV' computes the seasonal-latitudinal variation of density **
C** in the lower thermosphere in accordance with L. Jacchia, SAO 332, 1971.  **
C** This affects the densities between 90 and 170 km.  This subroutine need  **
C** not be called for densities above 170 km, because no effect is observed. **
C**                                                      **
C** The variation should be computed after the calculation of density due to **
C** temperature variations and the density ( DEN ) must be in the form of a  **
C** base 10 log.  No adjustments are made to the temperature or constituent  **
C** number densities in the region affected by this variation.        **
C**                                                   **
C**              DEN   = density (log10)                **
C**              ALT   = altitude (km)                 **
C**              XLAT  = latitude (rad)                **
C**              DAY   = day number                    **
C**                                                   **
C*********************************************************************
**

C** initialize density (DEN) = 0.0
C
         DEN = 0.0
C
C** check if altitude exceeds 170 km
C
         IF ( ALT. GT. 170. ) RETURN
C
C** compute density change in lower thermosphere
C
         Z = ALT - 90.
```

A27

```
          X = -0.0013 * Z * Z
          Y = 0.0172 * DAY + 1.72
          P = SIN (Y)
          SP = ( SIN (XLAT) ) **2
          S = 0.014 * Z * EXP (X)
          D = S * P * SP
C
C** check to compute absolute value of 'XLAT'
C
          IF ( XLAT. LT. 0. ) D = -D
          DEN = D

          RETURN
          END
```

```
          SUBROUTINE SLVH ( DEN , DENHE , XLAT , SDA )
```

```
C*************************************************************************
**
C** Subroutine 'SLVH' computes the seasonal-latitudinal varaition of the    **
C** helium number density according to L. Jacchia, SAO 332, 1971.  This     **
C** correction is not important below about 500 km.                    **
C**                                        **
C**           DEN  = density (log10)                  **
C**           DENHE = helium number density (log10)         **
C**           XLAT  = latitude (rad)               **
C**           SDA   = solar declination angle (rad)        **
C*************************************************************************
**
```

```
          D0 = 10. ** DENHE
          A = ABS ( 0.65 * ( SDA / 0.40909079 ) )

          B  = 0.5 * XLAT
C
C** Check to compute absolute value of 'B'
C
          IF ( SDA. LT. 0. ) B = -B
C
C** compute X, Y, DHE and DENHE
C
          X  = 0.7854 - B
          Y  = ( SIN (X) ) ** 3
          DHE= A * ( Y - 0.35356 )
          DENHE = DENHE + DHE
C
```

C** compute helium number density change
C

```
        D1 = 10. ** DENHE
        DEL= D1 - D0
        RHO= 10. ** DEN
        DRHO = ( 6.646E-24 ) * DEL
        RHO = RHO + DRHO
        DEN = ALOG10 (RHO)


        RETURN
        END
```

SUBROUTINE FAIR5 (DHEL1 ,DHEL2 ,DLG1 ,DLG2 ,IH ,FDHEL ,FDLG)

```
C**********************************************************************
C** This subroutine fairs between the region above 500 km, which invokes the **
C** seasonal-latitudinal variation of the helium number density ( subroutine **
C** SLVH ), and the region below, which does not invoke any seasonal-    **
C** latitudinal variation at all.                           **
C**                                           **
C** INPUTS:  DHEL1 = helium number density before invoking SLVH          **
C**          DHEL2 = helium number density after invoking SLVH        **
C**          DLG1  = total density before invoking SLVH            **
C**          DLG2  = total density after invoking SLVH            **
C**          IH    = height ( km ) -- INTEGER                **
C**          IBFH  = base fairing height ( km ) -- INTEGER           **
C** OUTPUTS: FDHEL = faired helium number density              **
C**          FDLG  = faired total density                **
C**                                           **
C** Written by Bill Jeffries, CSC, Huntsville, AL.             **
C**          ph. (205) 830-1000, x311                  **
C**********************************************************************
```

```
        DIMENSION CZ ( 6 )

        PARAMETER IBFH = 440
        DATA CZ / 1.0, 0.9045085, 0.6545085, 0.3454915, 0.0954915, 0.0 /
C Height index
        I = ( IH - IBFH ) /10 + 1
C Non-SLVH fairing coefficient
        CZI = CZ ( I )
C SLVH fairing coefficient
        SZI = 1.0 - CZI
C Faired density
        FDLG = ( DLG1 * CZI ) + ( DLG2 * SZI )
C Faired helium number density
        FDHEL = ( DHEL1 * CZI ) + ( DHEL2 * SZI )


        RETURN
        END
```

APPENDIX B

USER'S SOFTWARE IMPLEMENTATION GUIDE
AND
PROGRAM LISTINGS

A SIMULATION OF SMALL-SCALE
THERMOSPHERIC DENSITY
VARIATIONS FOR ENGINEERING
APPLICATIONS
(For orbital inclinations $\leq 40^{\circ}$)

USER'S SOFTWARE IMPLEMENTATION GUIDE
FOR ORBITAL INCLINATIONS LESS THAN OR EQUAL TO 40°

This section describes how to use the wave simulation model. In this section we employ the following conventions: subroutine and common blocks are written using all upper case letters, while only the first character of a variable name is written using upper case letters. There are several subroutines employed in the model. Their names, order of execution and functions are provided in Table B1, while their input and corresponding output are provded in Table B2. They are also included in Program Listing on p. B9.

Table B1. Subroutines, Order of Execution and Functions

Subroutine Name	Input Parameters	Output Parameters
SETWAVES	Pickbins: user input F_{13}bin and $A_p$bin (if Pickbins true): user input Iseed0: user input Warnings: user input	Iseed0: through argument list F_{13}bin, $A_p$bin and Warnings: all through COMMON /WAVE/
GETWAVES	Metden, F_{13}in, $a_p$in, and Iseed0: all through argument list	Totden: through argument list
WAVES	$A_p$in, F_{13}in, Fixdstep, Dt, Tmax, Iseed, Reset, Pickbins, F_{13}bin, $A_p$bin, Warnings: all through argument list	Wave: through argument list
GAUSSD	N and Iseed: through argument list	Noise: through argument list
RAN1	Idum (seed): through argument list	Ran1: function subprogram value returned

These subroutines have been designed to be used in conjunction with both the MET model and an orbit generator, although in principle they will work with anu empirical thermospheric density model. The subroutine WAVES is called and controlled by the subroutine GETWAVES. The input and output of WAVES are thoroughly described in the comments accompanying that subroutine, and will not need further explanation here. The subroutine GAUSSD and the function

subprogram RAN1, that together generate an array of Gaussian
distributed random numbers, need no further explanation here
except that the seed must be a large odd integer. The ref-
erences from which these portable and reliable codes were
obtained is given in the comments section of each of their
program codes.

Table B2. Subroutines and Their Input and Output.

Subroutine Name	Execution Order and Call Type	Function
SETWAVES	1 Single call	Performs routine setup. Requires user input for option to explicitly select F_{13} bins, the initial random number seed, and the option to suppress warning messages.
GETWAVES	2 Multiple call	Handles bookkeeping for WAVES subroutine. Inputs through argument list. Calls WAVES subroutine.
WAVES	3 Multiple call	Computes array of stochastic wave amplitudes. Inputs through argument list. Calls GAUSSD subroutine.
GAUSSD	4 Multiple call	Computes array of Gaussian random numbers. Inputs through argument list. Calls RAN1 subroutine
RAN1	5 Multiple call	Computes a single uniform random deviate between 0 and 1. Inputs through argument list.

Correct implementation of the set of subroutines will
require some additional comments regarding the subroutines
SETWAVES and GETWAVES. These are now briefly described.

Subroutine SETWAVES

The subroutine SETWAVES must be called initially to set
some of the inputs. If the decisions made during the execu-
tion of SETWAVES are adhered to for the entire simulation or
sets of simulations , SETWAVES need not be called again. In
this case Iseed), Pickbins and Warnings, as well as F_{13}bin
and $A_p$bin if Pickbins is true, will all remain unchanged.
If, however, Pickbins or Warnings need to be changed, then
SETWAVES will need to be recalled. In practice it is un-
likely, for example, that any user would want to run a simu-

lation first without suppressing warning statements and then later suppressing them. It would seem equally unlikely that during the course of a set of simulations any user would need to change from using a predetermined set of a_p and F_{13} values to using predetermined a_p and F_{13} *bin* values.

Subroutine GETWAVES

The subroutine GETWAVES needs to be called directly after the total neutral mass density, that has been calculated from the mean-state thermospheric model (e.g.,MET), has been returned to the orbit generator program. GETWAVES must be called every time that a new density value has been returned from the mean-state density model.

The first input parameter in the arument list of GETWAVES is Metden, the mean-state total neutral mass density output from the standard MET mode. An output Metden is modified by the addition of a stochastic component, as described earlier in the report. The other parameters in the argument list are F_{13}in, $A_p$in and Iseed0. As described previously, if SETWAVES is called once only, Iseed0 will remain unchanged. the input parameters F_{13}in and $A_p$in must be available within the orbit generator program whenever Pickbins is false so that they can be properly input to GETWAVES(in that case WAVES calculates the bins internally using the F_{13}in and $A_p$in values). If Pickbins is true, so that the bins are explicitly user-selected, F_{13}in and $A_p$in are not used at all.

Other parameters are passed to GETWAVES through a Common block(WAVE). The Common block must be included within the orbit generatore program *exactly* as specified in GETWAVES. The parameters in this Common block include the four logical constants Fixdstep, Reset, Pickbins and Warnings. The last three of these have their values set in SETWAVES(as previously discussed). The first logical constant, Fixdstep, must be set within the orbit generator program. Its setup belongs there, and not within SETWAVES, because the orbit generator program usually handles the time step, dt. For a single simulation, Fixdstep must remain constant. It can be changed only when Reset is true.

The remaining parameters in the Common block are Dt, Tmax, Iseed, F_{13}bin and $A_p$bin. As already discussed, F_{13}bin and $A_p$bin may have been previously set in SETWAVES. The time step, Dt, and the simulation time, Tmax, are set within the

orbit generator program. If Reset is true during the course of a simulation, the random number seed, Iseed, is reset by adding the integer 2 to its previous value. This method will ensure that Iseed remains large and odd while not interrrupting program execution(as would occur is a new value of Iseed0 was required to be manually input from a terminal by the user)>

A single simulation should not be allowed to exceed Tmax. If the elapsed time exceeds Tmax, Reset is reset to true, Elaptime is reset to zero, the random number seed, Iseed, is updated and a new simulation(which may follow the previous one) is initiated. Similarly, a new simulation is initiated if Pickbins is false and the a_p value changes. We don't expect F_{13} to change during the course of these relatively short duration simulations(less than a day), so no option exists for that remote possibility. If Tmax exceeds three hours and Pickbins is false, then it is likely that a_p will change because it is a three-hour index. Therefore, the simulations are not allowed to exceed three hours. If the user inputs a value of Tmax that is greater than three hours, the subroutine WAVES will reset that value to three hours and issue a warning(if Warnings is true).

Modifications Required In User-Supplied Orbit Generator Progam

The five following parameters need to be set by the user directly before calling GETWAVES within the user-written orbit generator program:
- a)   Dt   (set once only if using a fixed time step}
- b)   Fixdstep   (set once only)
- c) $F_{13}in$ (set once only)
- d) $A_p in$ (set once only)
- e)   Tmax   (set once only)_

The subroutines that need to be called from within the user-written orbit generator program and their calling sequence are:
- a)   Thermospheric density model(e.g.,MET)
          then immediately call next routine once only.
- b)   SETWAVES
          then immediately call next routine multiple times
- c)   GETWAVES

PROGRAM LISTINGS

The following pages contain the program listings for the subroutines SETWAVES, GETWAVES and GAUSSD and the function subprogram RAN1. The coding for this software is written in FORTRAN 77. It is not machine specific and should, therefore, be portable between different machines.

```
      SUBROUTINE SETWAVES (ISEED0)

      IMPLICIT      NONE
      INTEGER       APBIN, F13BIN, ISEED, ISEED0
      REAL          DT, TMAX
      CHARACTER*1   ANSWER
      LOGICAL       PICKBINS, WARNINGS, FIXDSTEP, RESET

      COMMON / WAVE / FIXDSTEP, DT, TMAX, ISEED, RESET, PICKBINS,
     .                F13BIN, APBIN, WARNINGS

C  Set option for explicitly selecting low or high F10.7 and ap bins for
C  wave simulator. The two F10.7 bins relate to the 13-month smoothed
C  value of F10.7 lying below 138 (low bin) or above 138 (high bin). The
C  two ap bins correspond to ap values below the 50th percentile (the
C  low ap bin for ap values less than or equal to 7) and those above the
C  50th percentile (the high ap bin for ap greater than 7). If the option
C  of explicitly selecting these bins is rejected, the bins are chosen
C  according to the values of 3-hourly ap and the 162-day mean F10.7
C  (assumed to approximate the 13-month smoothed F10.7) that are input
C  to the MET model. The option is particularly important if 13-month
C  smoothed values of predicted ap values are employed in simulations,
C  as they generally have values that lay solely in the **** ap bin.
      PRINT *,'Explicitly select F10 and ap bins for wave simulator?'
      PRINT *,'(Y/N)'
1     FORMAT (A1)
      READ 1, ANSWER
      IF (ANSWER.EQ.'Y'.OR.ANSWER.EQ.'y')  THEN
         PICKBINS = .TRUE.
         PRINT *,'Low (=1) or high (=2) 13-month F10.7 bin?'
         READ *, F13BIN
         PRINT *,'Low (=1) or high (=2) ap bin?'
         READ *, APBIN
      ELSE
         PICKBINS = .FALSE.
      END IF

C  An initial seed for the random number generator used in the wave
C  simulator is required. For subsequent simulations the seed is
C  internally incremented by 2.
      PRINT *,'Set initial wave simulator seed (large odd integer)'
      READ *, ISEED0

C  Warnings will be typed from the wave simulator if certain parameters
C  are incorrectly specified. For an "expert" user these can be
C  suppressed by setting the warnings option to false.
      PRINT *,'Suppress warning or error messages? (Y/N)'
      READ 1, ANSWER
      IF (ANSWER.EQ.'Y'.OR.ANSWER.EQ.'y')  THEN
         WARNINGS = .FALSE.
      ELSE
         WARNINGS = .TRUE.
      END IF

C  End setup for wave simulator


      RETURN
      END
```

```fortran
      SUBROUTINE GETWAVES (METDEN, F13IN, APIN, ISEED0, TOTDEN)
      IMPLICIT  NONE
      INTEGER   JSTEP, NRESET, ISEED, ISEED0, IWAVE, F13BIN, APBIN
      REAL      METDEN, F13IN, APIN, TOTDEN, DT, TIMESTEP(720),
     .          ELAPTIME, TMAX, OLDAP, WAVE(720)
      LOGICAL   FIXDSTEP, RESET, PICKBINS, WARNINGS

      COMMON / WAVE / FIXDSTEP, DT, TMAX, ISEED, RESET, PICKBINS,
     .                F13BIN, APBIN, WARNINGS

      IF (ELAPTIME.GT.TMAX .OR. (.NOT.RESET .AND..NOT.PICKBINS
     .    .AND. OLDAP.NE.APIN))  RESET = .TRUE.
      IF (RESET)  THEN
        NRESET = NRESET+1
        ISEED  = ISEED0+2*NRESET       ! this keeps seed odd
        OLDAP  = APIN
        ELAPTIME = 0.0
      END IF

Note that WAVES is called once only for fixed time steps
      IF (RESET.OR..NOT.FIXDSTEP)  THEN
        IWAVE = 0
        CALL WAVES (APIN, F13IN, FIXDSTEP, DT, TMAX, ISEED, RESET,
     .              PICKBINS, F13BIN, APBIN, WARNINGS, WAVE)
      END IF
      IWAVE    = IWAVE+1
      ELAPTIME = ELAPTIME+DT
      TOTDEN   = (1.0+WAVE(IWAVE))*METDEN

      RETURN
      END
```

SUBROUTINE WAVES (APIN, F13IN, FIXDSTEP, DT, TMAX, ISEED, RESET,
PICKBINS, F13BIN, APBIN, WARNINGS, WAVE)

This subroutine/model is based on the analysis of AEE NACE data (see
accompanying report by Michael P. Hickey).
It is valid for low-inclination, approximately circular orbits
between about 250 and 500 km altitude. The time resolution of the
original data and for this model is 15.0 seconds. The perturbations
output by this subroutine are used to modify the total mass density
of the MET model and are intended for simulations related to
atmospheric drag effects (e.g., guidance, navigation and control).

INPUTS:

APIN	is the same ap value that is input to MET model;
F13IN	is the 13-month smooth value of monthly mean F10.7;
FIXDSTEP	tells subroutine if time steps will be constant or if they will vary during the course of simulation;
DT	is either the fixed time step if FIXDSTEP is true or is the next time step if FIXDSTEP is false. In either case, the minimum allowable value of DT is 15 seconds (the time resolution of original data). If the input value of DT is less than 15 s it is automatically reset to 15.0;
TMAX	is the maximum time duration (sec) of simulation. It is advised that TMAX not exceed 3 hours (the time resolution of the ap index);
ISEED	is a large odd integer for random number generator;
RESET	tells subroutine to go back to setup mode if it is true. It should be set false on every subsequent call to WAVES within a single simulation. It should be set to true every three hours of simulation or whenever ap or F13 vary;
PICKBINS	tells subroutine to override APIN and F13IN inputs and explicitly select required AP and F13 bins;
F13BIN	is the explicitly selected F13 bin when PICKBINS is true;
APBIN	is the explicitly selected Ap bin when PICKBINS is true;
WARNINGS	tells subroutine to type all warnings that occur when either invalid input parameters are reset internally or when WAVES should be recalled in setup mode. If WARNINGS is false, no warnings are typed to the screen, although the user should be aware that internal resetting may be occuring if the subroutine is being misused.

OUTPUT:

WAVE	is the wave percentage amplitude expressed as a decimal. WAVE is an array dependant on FIXDSTEP. For FIXDSTEP false, only the first element of WAVE is nonzero; subroutine WAVES is then called for every time step to determine WAVE(1). For FIXDSTEP true, WAVE has 1+TMAX/15.0 elements for DT less than or equal to 15s and 1+TMAX/DT elements for DT greater than 15s. Subroutine WAVES is then called once only to determine WAVE(i), i=1, 1+min(DT,15.0).

Additional Subroutines Employed: GAUSSD and RAN1 (random # generators)

```fortran
      IMPLICIT   NONE
      INTEGER    ISEED, F13BIN, APBIN, N, K, I
      REAL       TMAX,
     .           APIN, F13IN, DT, WAVE(1+TMAX/DT), NOISE(721), PHI,
     .           STNDEV, AMP(721), RATIO, ELTIME, PHI1, PHI2,
     .           MARKOV(2), SCALE(2,2), AR2(4)
      LOGICAL    FIXDSTEP, SETUP, RESET, PICKBINS, WARNINGS,
     .           STOPFLAG
      DATA       MARKOV / 0.8531, 0.8324 /
      DATA       AR2    / 0.4672, 0.2841, 0.4902, 0.3046 /
      DATA       SCALE  / 2.00E-2, 1.80E-2, 1.91E-2, 1.92E-2 /
      DATA       SETUP    / .TRUE. /

      IF (RESET) SETUP = .TRUE.

C Check that DT and TMAX are both nonzero. Type fatal error message
C and terminate program execution if either one of them is zero.
      STOPFLAG = .FALSE.
      IF (DT.EQ.0.) THEN
        PRINT 1
        STOPFLAG = .TRUE.
      ELSE IF (TMAX.EQ.0.) THEN
        PRINT 2
        STOPFLAG = .TRUE.
      END IF
      IF (STOPFLAG) STOP

C Further check that DT is no smaller than 15 sec and that TMAX is no
C greater than 3 hrs. If they are, reset them to their nominal values.
      IF (DT.LT.15.0)  THEN
        DT = 15.0
        IF (WARNINGS) PRINT 3
      END IF
      IF (TMAX.GT.1.08E4)  THEN
        TMAX = 1.08E4
        IF (WARNINGS) PRINT 4
      END IF

C Commence setting F13 & ap bins, white noise array and wave amplitude
C array on a constant 15-sec time grid.
      IF (SETUP)  THEN

        SETUP = .FALSE.
        RESET = .FALSE.

C If PICKBINS is true, then ap and F13 bins are directly input.
C Otherwise, they are calculated from the input values of ap and F13.
        IF (.NOT.PICKBINS)  THEN
          IF (APIN.LE.7.0)  THEN
            APBIN = 1
          ELSE
            APBIN = 2
          END IF
          IF (F13IN.LE.138.0)  THEN
            F13BIN = 1
          ELSE
            F13BIN = 2
          END IF
        END IF
      END IF
```

B10

```
:   Set phi and estimated white noise standard deviation of process
          IF (F13BIN.EQ.1)  THEN
             PHI      = MARKOV (APBIN)
          ELSE
             PHI1     = AR2 (2*APBIN-1)
             PHI2     = AR2 (2*APBIN)
          END IF
          STNDEV   = SCALE  (F13BIN, APBIN)

          N = 1+TMAX/15.0
:   Generate white noise of zero mean and unit standard deviation
          CALL GAUSSD (NOISE, N, ISEED)

:   Generate array of wave amplitudes with 15-sec time resolution after
:   first initializing array to zero.
          DO  I = 1, N
             AMP(I) = 0.0
          END DO
          AMP(1) = NOISE(1)
          IF (F13BIN.EQ.1)  THEN
             DO  I = 2, N
                AMP(I) = PHI*AMP(I-1) + NOISE(I)
             END DO
          ELSE
             AMP(2) = PHI1*AMP(1) + NOISE(2)
             DO  I = 3, N
                AMP(I) = PHI1*AMP(I-1) + PHI2*AMP(I-2) + NOISE(I)
             END DO
          END IF
     Now make variance fit data by increasing standard deviation from
_    unity to STNDEV.
          DO  I = 1, N
             AMP(I) = STNDEV*AMP(I)
          END DO

-   If FIXDSTEP is true transform from 15-s time resolution array to DT
    time resolution array...
          IF (FIXDSTEP)  THEN
             RATIO = DT/15.0
             DO  I = 1, 1+TMAX/DT
                K = 1+INT(RATIO*REAL(I-1))
                WAVE(I) = AMP(K)
             END DO

:   ...else keep track of elapsed time
          ELSE
             ELTIME  = DT
             K       = 1+INT(ELTIME/15.0)
             WAVE(1) = AMP(K)
          END IF
          RETURN

     Next section is not setup, and should only be executed for FIXDSTEP
     false.
          ELSE
             IF (.NOT.FIXDSTEP)  THEN
                ELTIME  = ELTIME+DT
                IF (ELTIME.GT.TMAX)  THEN
                   PRINT 5
                   WAVE(1) = 0.0                B11
```

```
        ELSE
          K           = 1+INT(ELTIME/15.0)
          WAVE(1) = AMP(K)
        END IF
        RETURN
      ELSE
        IF (WARNINGS) PRINT 6
      END IF
    END IF

1     FORMAT(' FATAL ERROR: DT is zero in WAVES')
2     FORMAT(' FATAL ERROR: TMAX is zero in WAVES')
3     FORMAT(' WARNING: DT has been increased to 15 seconds')
4     FORMAT(' WARNING: TMAX has been decreased to 3 hours')
5     FORMAT(' WARNING: Elapsed time > TMAX. WAVE now set to 0.0',/,
     '                   Recall WAVES in setup mode')
6     FORMAT(' WARNING: Fixed time step. Recall WAVES in setup mode')

    RETURN
    END
```

```
      SUBROUTINE GAUSSD (X, N, ix)
c     routine to produce vector of gaussian distributed random numbers
c     in X(n).  The series in X(n) will have approximate zero mean and
c     standard deviation of 1.0.  Polar Method algorithm taken from
c     p. 104, Seminumerical Algorithms,
c     vol 2 of The Art of Computer Programming by D. E. Knuth, 1969.
c
c     ix is random number seed, i.e., large odd integer.
c
      implicit none
      integer i, ix, iy, n
      REAL X(N), RAN1
      real s, v1, v2


      I = 0
      iy = ix

      do while (i .lt. n)     .

          V1 = 2.0*ran1(iy) - 1.0
          V2 = 2.0*ran1(iy) - 1.0

          S = V1*V1 + V2*V2

           if (s .lt. 1.0) then

                S = SQRT(-2.0*ALOG(S) / S)

                I = I + 1
                X(I) = V1 * S
                I = I + 1
                if (i .le. n) then
                    X(I) = V2 * S
                end if

           end if

      end do

      ix = iy
      RETURN

      END
```

```
      FUNCTION RAN1 (IDUM)
C  Returns a uniform random deviate between 0.0 and 1.0. Set IDUM to any
C  negative value to initialize or reinitialize the sequence.
C  Copied from Numerical Recipes, W.H. Press, B.P. Flannery,
C  S.A. Teukolsky & W.T. Vetterling, Cambridge Univ. Press, 1986, p.196
      IMPLICIT   NONE
      INTEGER    IFF, IX1, IX2, IX3, J, M1, M2, M3, IA1, IA2, IA3,
                 IC1, IC2, IC3, IDUM
      REAL       RM1, RM2, R, RAN1
      DIMENSION  R(97)
      PARAMETER  (M1=259200, IA1=7141, IC1=54773, RM1=1./M1)
      PARAMETER  (M2=134456, IA2=8121, IC2=28411, RM2=1./M2)
      PARAMETER  (M3=243000, IA3=4561, IC3=51349)
      DATA  IFF  /0/  ! as above, initialize on first call even if
                      ! IDUM is not negative
      IF (IDUM.LT.0 .OR. IFF.EQ.0)  THEN
         IFF=1
         IX1=MOD(IC1-IDUM,M1)                ! Seed the first routine,
         IX1=MOD(IA1*IX1+IC1,M1)
         IX2=MOD(IX1,M2)                     ! and use it to seed the second
         IX1=MOD(IA1*IX1+IC1,M1)
         IX3=MOD(IX1,M3)                     ! and third routines.
         DO  J = 1, 97                       ! Fill the table with sequential
            IX1=MOD(IA1*IX1+IC1,M1)          ! uniform deviates generated by
            IX2=MOD(IA2*IX2+IC2,M2)          ! the first two routines
            R(J)=(FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1      ! Low and high-
         END DO                             ! order pieces combined here.
         IDUM=1
      END IF
      IX1=MOD(IA1*IX1+IC1,M1) ! Except when initializing, this is
      IX2=MOD(IA2*IX2+IC2,M2) ! where we start. Generate the next
      IX3=MOD(IA3*IX3+IC3,M3) ! number for each sequence.
      J=1+(97*IX3)/M3                        ! Use the third sequence to get an
                                             ! integer between 1 and 97
      IF(J.GT.97 .OR. J.LT.1) PAUSE
      RAN1=R(J)                              ! return that table entry,
      R(J)=(FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1   ! and refill it.
      RETURN
      END
```

APPENDIX C

USER'S SOFTWARE IMPLEMENTATION GUIDE
AND
PROGRAM LISTINGS

AN ENGINEERING MODEL FOR THE SIMULATION OF SMALL-SCALE
THERMOSPHERIC DENSITY VARIATIONS FOR ORBITAL INCLINATIONS
GREATER THAN 40°

USER'S SOFTWARE IMPLEMENTATION GUIDE
FOR ORBITAL INCLINATIONS GREATER THAN 40$^\text{o}$

This section describes how to use the wave simulation model. In this section we employ the following conventions: subroutines and common blocks are written using all uppercase letters, while only the first character of a variable name is written using uppercase letters. There are several subroutines employed in the model. Their names, order of execution and functions are provided in Table C1, while their input and corresponding output are provided in Table C2.

Table C1. Subroutines, order of execution and functions.

Subroutine Name	Execution Order & call type	Function
SETWAVE2	1 Single call	Performs routine setup. Requires user input for option to explicitly select a_p bin, the initial random number seed, and the option to suppress warning messages.
GETWAVE2	2 Multiple call	Handles bookkeeping for WAVES subroutine. Inputs through argument list. Calls WAVES subroutine.
WAVES2	3 Multiple call	Computes stochastic wave amplitude. Inputs through argument list. Calls GAUSSD subroutine.
GAUSSD	4 Multiple call	Computes array of Gaussian random numbers. Inputs through argument list. Calls RAN1 subroutine
RAN1	5 Multiple call	Computes a single uniform random deviate between 0 and 1. Inputs through argument list.

These subroutines have been designed to be used in conjunction with both the MET model and an orbit generator, although in principle they will work with any empirical thermospheric density model. The subroutine WAVES2 is called and controlled by the subroutine GETWAVE2. The input and output of WAVES2 are thoroughly described in the comments accompanying that subroutine, and will not need further explanation here. The subroutine GAUSSD and the function subprogram RAN1, that together generate an

array of Gaussian distributed random numbers, need no further explanation here except that the seed must be a large odd integer. The references from which these portable and reliable codes were obtained is given in the comments section of each of their program codes.

Table C2. Subroutines and their input and output.

Subroutine Name	Input Parameters	Output Parameters
SETWAVE2	**Pickbins**: user input A_p**bin** (if Pickbins true): user input **Iseed0**: user input **Warnings**: user input	**Iseed0**: through argument list A_p**bin** and **Warnings**: all through COMMON /WAVE/
GETWAVE2	**Metden, latin, $a_p$in**, and **Iseed0**: all through argument list	**Totden**: through argument list
WAVES2	A_p**in, latin, Dt, Tmax, Iseed, Reset, Pickbins, $A_p$bin, Warnings**: all through argument list	**Wave**: through argument list
GAUSSD	**N** and **Iseed**: through argument list	**Noise**: through argument list
RAN1	**Idum** (seed): through argument list	**Ran1**: function subprogram value returned

Correct implementation of the set of subroutines will require some additional comments regarding the subroutines SETWAVE2 and GETWAVE2. These are now briefly described.

Subroutine SETWAVE2

The subroutine SETWAVE2 must be called initially to set some of the inputs. If the decisions made during execution of SETWAVE2 are adhered to for the entire simulation or sets of simulations, SETWAVE2 need not be called again. In this case Iseed0, Pickbins and Warnings, as well as $a_p$bin if Pickbins is true, will all remain unchanged. If, however, Pickbins or Warnings need to be changed, then SETWAVE2 will need to be recalled. In

practice it is unlikely, for example, that any user would want to run a simulation first without suppressing warning statements and then later suppressing them. It would seem equally unlikely that during the course of a set of simulations any user would need to change from using a predetermined a_p value to using a predetermined a_p *bin* values.

Subroutine GETWAVE2

The subroutine GETWAVE2 needs to be called directly after the total neutral mass density, that has been calculated from the mean-state thermospheric model (e.g., MET), has been returned to the orbit generator program. GETWAVE2 must be called every time that a new density value has been returned from the mean-state density model.

The first input parameter in the argument list of GETWAVE2 is Metden, the mean-state total neutral mass density output from the standard MET model. On output Metden is modified by the addition of a stochastic component, as described earlier in this report. The other parameters in the argument list are latin, $a_p$in and Iseed0. As described previously, if SETWAVE2 is called once only, Iseed0 will remain unchanged. The input parameters latin and $a_p$in must be available within the orbit generator program whenever Pickbins is false so that they can be properly input to GETWAVE2 (in that case WAVES2 calculates the bins internally using the latin and $a_p$in values). If Pickbins is true, so that the a_p bin is explicitly user-selected, $a_p$in is not used at all. We strongly emphasize that a latitude bin cannot be explicitly selected even if Pickbins is true, because latitude will always vary (and the latitude bins will too) around an orbit for which this simulator was designed (i.e., for inclinations grater than about 45°).

Other parameters are passed to GETWAVE2 through a Common block (WAVE). This Common block must be included within the orbit generator program *exactly* as specified in GETWAVE2. The parameters in this Common block include the three logical constants Reset, Pickbins and Warnings. They have their values set in SETWAVE2 (as previously discussed).

The remaining parameters in the Common block are Dt, Tmax, Iseed and $a_p$bin. As already discussed, $a_p$bin may have been previously set in SETWAVE2. The time step, Dt, and the simulation time, Tmax, are set within the orbit generator program. If Reset is reset

C4

to true during the course of a simulation, the random number seed, Iseed, is reset by add-ing the integer 2 to its previous value. This method will ensure that Iseed remains large and odd while not interrupting program execution (as would occur if a new value of Iseed0 was required to be manually input from a terminal by the user).

A single simulation should not be allowed to exceed Tmax. If the elapsed time exceeds Tmax, Reset is reset to true, Elaptime is reset to zero, the random number seed (Iseed) is updated, and a new simulation (which may follow the previous one) is initiated. Similarly, a new simulation is initiated if Pickbins is false and the a_p value changes. If Tmax exceeds three hours and Pickbins is false, then it is likely that a_p will change because it is a three hour index. Therefore the simulations are not allowed to exceed three hours. If the user inputs a value of Tmax that is greater than three hours, the subroutine WAVES2 will reset that value to three hours and issue a warning (if Warnings is true).

Modifications Required in User-Supplied Orbit Generator Program

The four following parameters need to be set by the user directly before calling GET-WAVES within the user-written orbit generator program:

a) Dt   (set once only if using a fixed time step)

b) latin   (set on every call)

c) $a_p$in   (set once only)

d) Tmax   (set once only)

The subroutines that need to be called from within the user-written orbit generator pro-gram and their calling sequence are

a) <u>Thermospheric density model</u> (e.g., MET)

  then immediately call next routine **once only**

b) <u>SETWAVE2</u>

  then immediately call next routine **multiple times**

c) <u>GETWAVE2</u>

PROGRAM LISTINGS

The following pages contain the program listings for the subroutines SETWAVE2, GETWAVE2, WAVES2 and GAUSSD and the function subprogram RAN1. The coding for this software is written in FORTRAN 77. It is not machine specific and should therefore be portable between different machines.

```fortran
      SUBROUTINE SETWAVE2 (ISEED0)

      IMPLICIT    NONE
      INTEGER     APBIN, LATBIN, ISEED, ISEED0
      REAL        DT, TMAX
      CHARACTER*1 ANSWER
      LOGICAL     PICKBINS, WARNINGS, RESET

      COMMON / WAVE / DT, TMAX, ISEED, RESET, PICKBINS,
     >                APBIN, WARNINGS

C...Set option for explicitly selecting low or high ap bin for wave
C...simulator.
C...Two ap bins correspond to ap values below the 50th percentile (the
C...low ap bin for ap values less than or equal to 7) and those above the
C...50th percentile (the high ap bin for ap greater than 7). If the option
C...of explicitly selecting these bins is rejected, the bin is chosen
C...according to the values of 3-hourly ap that are input to the MET model.
      PRINT *,'Explicitly select ap bin for wave simulator?'
      PRINT *,'(Y/N)'
1     FORMAT (A1)
      READ 1, ANSWER
      IF (ANSWER.EQ.'Y'.OR.ANSWER.EQ.'y')  THEN
        PICKBINS = .TRUE.
        PRINT *,'Low (=1) or high (=2) ap bin?'
        READ *, APBIN
      ELSE
        PICKBINS = .FALSE.
      END IF

C...An initial seed for the random number generator used in the wave
C...simulator is required. For subsequent simulations the seed is
C...internally incremented by 2.
      PRINT *,'Set initial wave simulator seed (large odd integer)'
      READ *, ISEED0

C...Warnings will be typed from the wave simulator if certain parameters
C...are incorrectly specified. For an "expert" user these can be
C...suppressed by setting the warnings option to false.
      PRINT *,'Suppress warning or error messages? (Y/N)'
      READ 1, ANSWER
      IF (ANSWER.EQ.'Y'.OR.ANSWER.EQ.'y')  THEN
        WARNINGS = .FALSE.
      ELSE
        WARNINGS = .TRUE.
      END IF

C...End setup for wave simulator

      RETURN
      END
```

```fortran
      SUBROUTINE GETWAVE2 (METDEN, LATIN, APIN, ISEED0, TOTDEN)
      IMPLICIT    NONE
      INTEGER     JSTEP, NRESET, ISEED, ISEED0, APBIN, LATBIN,
     >            OLDLATBIN
      REAL        METDEN, LATIN, APIN, TOTDEN, DT, TIMESTEP(720),
     >            ELAPTIME, TMAX, OLDAP, WAVE
      LOGICAL     RESET, PICKBINS, WARNINGS

      COMMON / WAVE / DT, TMAX, ISEED, RESET, PICKBINS,
     >                APBIN, WARNINGS

      IF (ELAPTIME.GT.TMAX .OR. (.NOT.RESET .AND..NOT.PICKBINS
     >    .AND. OLDAP.NE.APIN))  RESET = .TRUE.
      IF (ABS(LATIN).GT.40.0)  THEN
        LATBIN = 2
      ELSE
        LATBIN = 1
      END IF
      IF (LATBIN.NE.OLDLATBIN) RESET = .TRUE.

      IF (RESET)  THEN
        NRESET = NRESET+1
        ISEED  = ISEED0+2*NRESET       ! this keeps seed odd
        OLDAP  = APIN
        OLDLATBIN = LATBIN
        ELAPTIME = 0.0
      END IF

C...Note that WAVES is called every time
      CALL WAVES2 (APIN, LATIN, DT, TMAX, ISEED, RESET,
     >             PICKBINS, APBIN, WARNINGS, WAVE)
      ELAPTIME = ELAPTIME+DT
      TOTDEN   = (1.0+WAVE)*METDEN

      RETURN
      END
```

```
              SUBROUTINE WAVES2 (APIN, LATIN, DT, TMAX, ISEED, RESET,
       >                         PICKBINS, APBIN, WARNINGS, WAVE)

C...This subroutine/model is based on the analysis of AEC NATE data (see
C...accompanying report by Michael P. Hickey, 1995).
C...It is valid only for inclinations greater than about 45 degrees
C...and for approximately circular orbits between about 200 and 500 km
C...The time resolution of the original data and for this model is
C...15.0 seconds. The perturbations output by this subroutine are used
C...to modify the total mass density of the MET model and are intended
C...for simulations related to atmospheric drag effects (e.g., guidance,
C...navigation and control).

C...NPUTS:   APIN      is the same ap value that is input to MET model;
C...         LATIN     is latitude;
C...         DT        is the next time step, which must be at least
C...                   15 seconds (the time resolution of original data).
C...                   If the input value of DT is less than 15 s it is
C...                   automatically reset to 15.0;
C...         TMAX      is the maximum time duration (sec) of simulation.
C...                   It is advised that TMAX not exceed 3 hours (the
C...                   time resolution of the ap index);
C...         ISEED     is a large odd integer for random number generator;
C...         RESET     tells subroutine to go back to setup mode if it is
C...                   true. It should be reset when the ap  bins varies;
C...         PICKBINS  tells subroutine to override APIN inputs and
C...                   explicitly select required ap bin;
C...         APBIN     is the explicitly selected ap bin when PICKBINS is
C...                   true;
C...         WARNINGS  tells subroutine to type all warnings that occur
C...                   when either invalid input parameters are reset
C...                   internally or when WAVES2 should be recalled in
C...                   setup mode. If WARNINGS is false, no warnings are
C...                   typed to the screen, although the user should be
C...                   aware that internal resetting may be occuring if
C...                   the subroutine is being misused.

C...OUTPUT:   WAVE     is the wave percentage amplitude expressed as a
C...                   decimal. WAVE is a single dependant variable.
C...                   Subroutine WAVES2 is called every time step.

C...Additional Subroutines Employed: GAUSSD and RAN1 (random # generators)
```

```fortran
          IMPLICIT      NONE
          INTEGER       ISEED, LATBIN, APBIN, N, K, I, IGRID, JGRID
          REAL          TMAX,
     >                  APIN, LATIN, DT, A, A1, A2, WAVE, NOISE(721), PHI,
     >                  STNDEV, RATIO, ELTIME, PHI1, PHI2,
     >                  SCALE(2,2), AR2(8)
          LOGICAL       FIXDSTEP, SETUP, RESET, PICKBINS, WARNINGS,
     >                  STOPFLAG
C...AE-C parameters:
          DATA          AR2      / 1.1792, -0.24832, 1.2764, -0.33955,
     >                             1.4367, -0.51164, 1.3461, -0.42737 /
          DATA          SCALE    / 1.385E-2, 2.093E-2, 1.171E-2, 2.440E-2 /
          DATA          SETUP      / .TRUE. /

          IF (RESET) SETUP = .TRUE.

C...Check that DT and TMAX are both nonzero. Type fatal error message
C...and terminate program execution if either one of them is zero.
          STOPFLAG = .FALSE.
          IF (DT.EQ.0.) THEN
            PRINT 1
            STOPFLAG = .TRUE.
          ELSE IF (TMAX.EQ.0.) THEN
            PRINT 2
            STOPFLAG = .TRUE.
          END IF
          IF (STOPFLAG) STOP

C...Further check that DT is no smaller than 15 sec and that TMAX is no
C...greater than 3 hrs. If they are, reset them to their nominal values.
          IF (DT.LT.15.0)  THEN
            DT = 15.0
            IF (WARNINGS) PRINT 3
          END IF
          IF (TMAX.GT.1.08E4)  THEN
            TMAX = 1.08E4
            IF (WARNINGS) PRINT 4
          END IF

C...Commence setting latitide & ap bins, white noise array on a constant
C...15-sec time grid.
          IF (SETUP)  THEN

            SETUP = .FALSE.
            RESET = .FALSE.


            N = 1+TMAX/15.0
C...Generate white noise of zero mean and unit standard deviation
            CALL GAUSSD (NOISE, N, ISEED)

C...PICKBINS = true will override the input value of ap.
            IF (.NOT.PICKBINS)  THEN
              IF (APIN.LE.7.0)  THEN
                APBIN = 1
              ELSE
                APBIN = 2
              END IF
            END IF
            IF (ABS(LATIN).LE.40.0)  THEN
```
C10

```
                LATBIN = 1
                PHI1 = AR2 (2*APBIN-1)
                PHI2 = AR2 (2*APBIN)
              ELSE
                LATBIN = 2
                PHI1 = AR2 (4+2*APBIN-1)
                PHI2 = AR2 (4+2*APBIN)
              END IF
              STNDEV = SCALE (LATBIN,APBIN)
            END IF

C...Wave model is valid on a fixed 15 second grid, but any variable
C...time step can be input. We must therefore keep track of the total
C...elapsed time, and at each (possibly variable) step determine the
C...corresponding position on the fixed 15 second grid.
C...Note that IGRID is the position on the fixed 15-sec grid, while
C...JGRID is the projected position calculated from the elapsed time.
C...For first time:
            IF (ELTIME.EQ.0.0)  THEN
              IGRID = 1
              JGRID = 1
              A = NOISE(1)
C...and for all subsequent times:
            ELSE
              JGRID = 1+INT(ELTIME/15.0)
              DO WHILE (IGRID.LT.JGRID)
                IGRID = IGRID+1
                A = PHI1*A1 + PHI2*A2 + NOISE(IGRID)
              END DO
            END IF
            A2 = A1
            A1 = A

C  Now make variance fit data by increasing standard deviation from
C  unity to STNDEV.
            WAVE = STNDEV*A

            IF (ELTIME.GT.TMAX)  THEN
              PRINT 5
              WAVE = 0.0
            END IF
            ELTIME = ELTIME+DT

1           FORMAT(' FATAL ERROR: DT is zero in WAVES2')
2           FORMAT(' FATAL ERROR: TMAX is zero in WAVES2')
3           FORMAT(' WARNING: DT has been increased to 15 seconds')
4           FORMAT(' WARNING: TMAX has been decreased to 3 hours')
5           FORMAT(' WARNING: Elapsed time > TMAX. WAVE now set to 0.0',/,
     >                 '         Recall WAVES2 in setup mode')

            RETURN
            END
```

```fortran
      SUBROUTINE GAUSSD (X, N, ix)
c     routine to produce vector of gaussian distributed random numbers
c     in X(n).  The series in X(n) will have approximate zero mean and
c     standard deviation of 1.0.  Polar Method algorithm taken from
c     p. 104, Seminumerical Algorithms,
c     vol 2 of The Art of Computer Programming by D. E. Knuth, 1969.
c
c     ix is random number seed, i.e., large odd integer.
c
      implicit none
      integer i, ix, iy, n
      REAL X(N), RAN1
      real s, v1, v2


      I = 0
      iy = ix

      do while (i .lt. n)

         V1 = 2.0*ran1(iy) - 1.0
         V2 = 2.0*ran1(iy) - 1.0

         S = V1*V1 + V2*V2

          if (s .lt. 1.0) then

              S = SQRT(-2.0*ALOG(S) / S)

              I = I + 1
              X(I) = V1 * S
              I = I + 1
              if (i .le. n) then
                  X(I) = V2 * S
              end if

          end if

      end do

      ix = iy
      RETURN

      END
```

C12

```fortran
      FUNCTION RAN1 (IDUM)
C  Returns a uniform random deviate between 0.0 and 1.0. Set IDUM to any
C  negative value to initialize or reinitialize the sequence.
C  Copied from Numerical Recipes, W.H. Press, B.P. Flannery,
C  S.A. Teukolsky & W.T. Vetterling, Cambridge Univ. Press, 1986, p.196
      IMPLICIT   NONE
      INTEGER    IFF, IX1, IX2, IX3, J, M1, M2, M3, IA1, IA2, IA3,
     >           IC1, IC2, IC3, IDUM
      REAL       RM1, RM2, R, RAN1
      DIMENSION  R(97)
      PARAMETER  (M1=259200, IA1=7141, IC1=54773, RM1=1./M1)
      PARAMETER  (M2=134456, IA2=8121, IC2=28411, RM2=1./M2)
      PARAMETER  (M3=243000, IA3=4561, IC3=51349)
      DATA  IFF  /0/  ! as above, initialize on first call even if
                      ! IDUM is not negative
      IF (IDUM.LT.0 .OR. IFF.EQ.0)  THEN
         IFF=1
         IX1=MOD(IC1-IDUM,M1)            ! Seed the first routine,
         IX1=MOD(IA1*IX1+IC1,M1)
         IX2=MOD(IX1,M2)                 ! and use it to seed the second
         IX1=MOD(IA1*IX1+IC1,M1)
         IX3=MOD(IX1,M3)                 ! and third routines.
         DO  J = 1, 97                   ! Fill the table with sequential
             IX1=MOD(IA1*IX1+IC1,M1)     ! uniform deviates generated by
             IX2=MOD(IA2*IX2+IC2,M2)     ! the first two routines
             R(J)=(FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1       ! Low and high-
         END DO                          ! order pieces combined here.
         IDUM=1
      END IF
      IX1=MOD(IA1*IX1+IC1,M1) ! Except when initializing, this is
      IX2=MOD(IA2*IX2+IC2,M2) ! where we start. Generate the next
      IX3=MOD(IA3*IX3+IC3,M3) ! number for each sequence.
      J=1+(97*IX3)/M3         ! Use the third sequence to get an
                             ! integer between 1 and 97
      IF(J.GT.97 .OR. J.LT.1) PAUSE
      RAN1=R(J)                          ! return that table entry,
      R(J)=(FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1    ! and refill it.
      RETURN
      END
```

APPENDIX D

USER'S SOFTWARE IMPLMENTATION GUIDE
AND
PROGRAM LISTINGS

THE MET MODEL
STATISTICAL ANALYSIS MODE
(MET-SAM)

USER'S SOFTWARE IMPLEMENTATION GUIDE

To use the statistical analysis mode(SAM) of the **MET** model select the appropriate exospheric temperature from the following tables and the altitude for which you desire the density to be computed. Enter these two values into the subroutine program code. To obtain an exospheric temperature value at a specific percentile level that is not in the table, linearly interpolate between the two nearest values. PFD is the percent frequency distribution.

Table 1. Cumulative Percent Frequency Distribution of Minimum Exospheric Temperature When 13-Month Smoothed 10.7-cm Solar Flux Is Between 66 and 246

PFD	CPF		PFD	CPF		PFD	CPF	
0.0256	0.02	567.00	2.59	75.21	987.00	0.16	99.37	1407.00
0.29	0.31	587.00	2.54	77.74	1007.00	0.17	99.54	1427.00
1.34	1.66	607.00	2.29	80.04	1027.00	0.13	99.67	1447.00
2.70	4.36	627.00	2.30	82.34	1047.00	0.09	99.76	1467.00
4.33	8.69	647.00	2.24	84.57	1067.00	0.06	99.82	1487.00
5.87	14.56	667.00	1.97	86.54	1087.00	0.05	99.88	1507.00
6.11	20.68	687.00	1.80	88.34	1107.00	0.05	99.92	1527.00
5.40	26.08	708.00	1.60	89.93	1127.00	0.03	99.95	1547.00
5.01	31.09	727.00	1.49	91.43	1147.00	0.02	99.97	1567.00
4.79	35.88	747.00	1.34	92.77	1167.00	0.01	99.98	1587.00
4.43	40.31	767.00	1.18	93.95	1187.00	0.01	99.98	1607.00
4.05	44.36	787.00	1.11	95.06	1207.00	0.01	99.99	1627.00
3.61	47.98	807.00	0.94	96.00	1227.00	0.00	99.99	1647.00
3.34	51.31	827.00	0.73	96.73	1247.00	0.01	100.00	1667.00
3.27	54.59	847.00	0.60	97.33	1267.00	0.00	100.00	1687.00
3.38	57.96	867.00	0.50	97.82	1287.00	0.00	100.00	1707.00
3.25	61.21	887.00	0.41	98.23	1307.00	0.00	100.00	1727.00
3.09	64.30	907.00	0.36	98.59	1327.00	0.00	100.00	1747.00
3.05	67.35	927.00	0.25	98.83	1347.00	0.00	100.00	1767.00
2.61	69.96	947.00	0.18	99.02	1367.00			
2.65	72.61	967.00	0.19	99.21	1387.00			

Table 2. Cumulative Percent Frequency Distribution of Mean Exospheric Temperature When 13-Month Smoothed 10.7-cm Solar Flux Is Between 66 and 246

PFD	CPF		PFD	CPF		PFD	CPF	
0.02	0.02	660.00	2.27	73.77	1120.00	0.16	99.19	1580.00
0.24	0.26	680.00	2.32	76.09	1140.00	0.16	99.35	1600.00
1.11	1.37	700.00	2.19	78.28	1160.00	0.16	99.51	1620.00
2.27	3.65	720.00	2.03	80.31	1180.00	0.13	99.64	1640.00
3.72	7.36	740.00	2.00	82.30	1200.00	0.09	99.74	1660.00
5.28	12.65	760.00	1.92	84.22	1220.00	0.06	99.79	1680.00
5.76	18.41	780.00	1.88	86.10	1240.00	0.06	99.85	1700.00
5.30	23.71	800.00	1.58	87.68	1260.00	0.04	99.90	1720.00
4.81	28.52	820.00	1.54	89.22	1280.00	0.04	99.94	1740.00
4.42	32.94	840.00	1.36	90.58	1300.00	0.02	99.96	1760.00
4.01	36.95	860.00	1.30	91.88	1320.00	0.01	99.97	1780.00
3.74	40.69	880.00	1.16	93.03	1340.00	0.01	99.98	1800.00
3.48	44.17	900.00	1.06	94.09	1360.00	0.01	99.99	1820.00
3.14	47.30	920.00	0.99	95.08	1380.00	0.00	99.99	1840.00
2.90	50.20	940.00	0.84	95.92	1400.00	0.01	100.00	1860.00
2.78	52.97	960.00	0.69	96.61	1420.00	0.00	100.00	1880.00
2.79	55.77	980.00	0.57	97.18	1440.00	0.00	100.00	1900.00
2.84	58.60	1000.00	0.48	97.65	1460.00	0.00	100.00	1920.00
2.73	61.33	1020.00	0.42	98.07	1480.00	0.00	100.00	1940.00
2.81	64.14	1040.00	0.31	98.39	1500.00	0.00	100.00	1960.00
2.70	66.84	1060.00	0.30	98.69	1520.00	0.00	100.00	1980.00
2.40	69.25	1080.00	0.19	98.87	1540.00			
2.25	71.50	1100.00	0.16	99.03	1560.00			

Table 3. Cumulative Percent Frequency Distribution of Maximum Exospheric Temperature When 13-Month Smoothed 10.7-cm Solar Flux Is Between 66 and 246

PFD	CPF		PFD	CPF		PFD	CPF	
0.01	0.01	753.00	1.96	72.69	1253.00	0.13	99.04	1753.00
0.22	0.23	773.00	2.01	74.71	1273.00	0.16	99.19	1773.00
0.93	1.16	793.00	2.03	76.74	1293.00	0.16	99.35	1793.00
1.97	3.13	813.00	2.03	78.77	1313.00	0.13	99.48	1813.00
3.15	6.28	833.00	1.80	80.57	1333.00	0.13	99.61	1833.00
4.74	11.02	853.00	1.71	82.28	1353.00	0.11	99.72	1853.00
5.45	16.47	873.00	1.65	83.93	1373.00	0.05	99.77	1873.00
5.22	21.69	893.00	1.66	85.60	1393.00	0.06	99.82	1893.00
4.60	26.29	913.00	1.54	87.14	1413.00	0.05	99.88	1913.00
4.21	30.50	933.00	1.41	88.54	1433.00	0.04	99.92	1933.00
3.79	34.29	953.00	1.30	89.84	1453.00	0.03	99.94	1953.00
3.51	37.80	973.00	1.23	91.07	1473.00	0.01	99.96	1973.00
3.23	41.20	993.00	1.11	92.18	1493.00	0.01	99.96	1993.00
2.91	43.94	1013.00	1.05	93.22	1513.00	0.02	99.98	2013.00
2.79	46.73	1033.00	0.97	94.19	1533.00	0.01	99.99	2033.00
2.62	49.53	1053.00	0.92	95.11	1553.00	0.01	99.99	2053.00
2.51	51.86	1073.00	0.77	95.89	1573.00	0.01	100.00	2073.00
2.34	54.20	1093.00	0.63	96.51	1593.00	0.00	100.00	2093.00
2.36	56.55	1113.00	0.50	0.50	1613.00	0.00	100.00	2113.00
2.50	59.06	1133.00	0.46	0.46	1633.00	0.00	100.00	2133.00
2.36	61.42	1153.00	0.43	97.90	1653.00	0.00	100.00	2153.00
2.50	63.92	1173.00	0.33	98.23	1673.00	0.00	100.00	2173.00
2.47	66.38	1193.00	0.29	98.52	1693.00	0.00	100.00	2183.00
2.24	68.62	1213.00	0.22	98.73	1713.00			
2.12	70.74	1233.00	0.17	98.91	1733.00			

Table 4. Cumulative Percent Frequency Distribution of Minimum Exospheric Temperature When 13-Month Smoothed 10.7-cm Solar Flux Is Between 66 and 102

PFD	CPF		PFD	CPF		PFD	CPF	
0.04	0.04	567.00	4.23	92.67	787.00	0.02	99.98	1007.00
0.74	0.78	587.00	2.85	95.53	807.00	0.01	99.98	1027.00
3.39	4.17	607.00	1.72	97.24	827.00	0.00	99.98	1047.00
6.80	10.98	627.00	1.13	98.37	847.00	0.01	99.99	1067.00
10.84	21.82	647.00	0.72	99.09	867.00	0.01	100.00	1087.00
14.65	36.47	667.00	0.35	99.44	887.00	0.00	100.00	1107.00
14.84	51.31	687.00	0.25	99.69	907.00	0.00	100.00	1127.00
12.25	63.56	707.00	0.13	99.82	927.00	0.00	100.00	1147.00
10.17	73.73	727.00	0.07	99.88	947.00	0.00	100.00	1167.00
8.36	82.09	747.00	0.06	99.94	967.00	0.00	100.00	1173.00
6.36	88.44	767.00	0.01	99.96	987.00			

Table 5. Cumulative Percent Frequency Distribution of Mean Exospheric Temperature When 13-Month Smoothed 10.7-cm Solar Flux Is Between 66 and 102

PFD	CDF		PFD	CDF		PFD	CDF	
0.04	0.04	660.00	5.03	90.32	880.00	0.03	99.96	1100.00
0.62	0.66	680.00	3.48	93.80	900.00	0.01	99.97	1120.00
2.81	3.47	700.00	2.25	96.05	920.00	0.01	99.98	1140.00
5.73	9.20	720.00	1.42	97.47	940.00	0.01	99.99	1160.00
9.33	18.54	740.00	1.04	98.50	960.00	0.00	99.99	1180.00
13.23	31.76	760.00	0.66	99.16	980.00	0.01	100.00	1200.00
14.35	46.12	780.00	0.29	99.45	1000.00	0.00	100.00	1220.00
12.76	58.87	800.00	0.22	99.67	1020.00	0.00	100.00	1240.00
10.80	69.67	820.00	0.14	99.82	1040.00	0.00	100.00	1260.00
8.82	78.49	840.00	0.08	99.90	1060.00	0.00	100.00	1278.00
6.80	85.29	860.00	0.04	99.94	1080.00			

Table 6. Cumulative Percent Frequency Distribution of Maximum Exospheric Temperature When 13-Month Smoothed 10.7-cm Solar Flux is Between 66 and 102

PFD	CPF		PFD	CPF		PFD	CPF	
0.04	0.04	753.00	5.70	87.60	973.00	0.04	99.94	1193.00
0.55	0.59	773.00	4.14	91.74	993.00	0.02	99.96	1213.00
2.53	2.94	793.00	2.71	94.45	1013.00	0.01	99.97	1233.00
4.97	7.91	813.00	1.98	96.43	1033.00	0.01	99.98	1253.00
7.92	15.82	833.00	1.25	97.68	1053.00	0.01	99.99	1273.00
11.89	27.71	853.00	0.87	98.56	1073.00	0.01	99.99	1293.00
13.69	41.40	873.00	0.59	99.15	1093.00	0.00	100.00	1313.00
12.95	54.35	893.00	0.32	99.47	1113.00	0.00	100.00	1333.00
10.90	65.25	913.00	0.18	99.64	1133.00	0.00	100.00	1353.00
9.32	74.57	933.00	0.17	99.81	1153.00	0.00	100.00	1373.00
7.33	81.90	953.00	0.09	99.90	1173.00	0.00	100.00	1383.00

Table 7. Cumulative Percent Frequency Distribution of Minimum Exospheric Temperature When 13-Month Smoothed 10.7 Solar Flux is Between 102 and 138

PFD	CPF		PFD	CPF		PFD	CPF	
0.03	0.03	615.00	4.99	86.61	915.00	0.00	99.93	1215.00
0.08	0.11	635.00	3.94	90.55	935.00	0.01	99.94	1235.00
0.22	0.33	655.00	2.93	93.48	955.00	0.01	99.95	1255.00
0.61	0.95	675.00	2.24	95.72	975.00	0.01	99.96	1275.00
1.86	2.81	695.00	1.46	97.17	995.00	0.01	99.97	1295.00
3.35	6.16	715.00	1.04	98.21	1015.00	0.01	99.98	1315.00
5.60	11.77	735.00	0.65	98.86	1035.00	0.00	99.98	1335.00
7.30	19.06	755.00	0.37	99.23	1055.00	0.01	99.99	1355.00
8.86	27.92	775.00	0.27	99.50	1075.00	0.00	99.99	1375.00
10.13	38.05	795.00	0.12	99.62	1095.00	0.00	99.99	1395.00
9.89	47.94	815.00	0.14	99.76	1115.00	0.00	99.99	1415.00
9.30	57.23	835.00	0.05	99.82	1135.00	0.00	99.99	1435.00
9.17	66.40	855.00	0.06	99.88	1155.00	0.01	100.00	1455.00
8.58	74.98	875.00	0.04	99.92	1175.00	0.00	100.00	1467.00
6.64	81.62	895.00	0.00	99.93	1195.00			

Table 8. Cumulative Percent Frequency Distribution of Mean Exospheric Temperature When 13-Month Smoothed 10.7-cm Solar Flux Is Between 102 and 138

PFD	CPF		PFD	CPF		PFD	CPF	
0.03	0.03	718.00	4.99	85.55	1038.00	0.00	99.93	1358.00
0.08	0.11	738.00	3.67	89.22	1058.00	0.01	99.94	1378.00
0.22	0.32	758.00	2.96	92.18	1078.00	0.01	99.95	1398.00
0.36	0.68	778.00	2.42	94.60	1098.00	0.01	99.96	1418.00
1.18	1.86	798.00	1.76	96.37	1118.00	0.00	99.96	1438.00
2.53	4.39	818.00	1.22	97.59	1138.00	0.01	99.98	1458.00
4.34	8.74	838.00	0.80	98.39	1158.00	0.00	99.98	1478.00
6.21	14.95	858.00	0.48	98.87	1178.00	0.01	99.99	1498.00
7.51	22.46	878.00	0.36	99.23	1198.00	0.00	99.99	1518.00
8.81	31.27	898.00	0.28	99.50	1218.00	0.00	99.99	1538.00
9.14	40.41	918.00	0.13	99.64	1238.00	0.00	99.99	1558.00
9.35	49.76	938.00	0.08	99.72	1258.00	0.00	99.99	1578.00
8.55	58.31	958.00	0.10	99.82	1278.00	0.01	100.00	1598.00
8.08	66.39	978.00	0.06	99.88	1298.00	0.00	100.00	1608.00
7.69	74.08	998.00	0.04	99.92	1318.00			
6.48	80.56	1018.00	0.00	99.93	1338.00			

Table 9. Cumulative Percent Frequency Distribution of Maximum Exospheric Temperature When 13-Month Smoothed 10.7-cm Solar Flux Is Between 102 and 138

PFD	CPF		PFD	CPF		PFD	CPF	
0.03	0.03	820.00	6.28	79.22	1140.00	0.04	99.91	1460.00
0.08	0.11	840.00	4.97	84.19	1160.00	0.02	99.93	1480.00
0.18	0.28	860.00	3.86	88.05	1180.00	0.00	99.93	1500.00
0.24	0.52	880.00	2.65	90.70	1200.00	0.01	99.94	1520.00
0.64	1.16	900.00	2.44	93.13	1220.00	0.01	99.95	1540.00
2.00	3.16	920.00	2.13	95.26	1240.00	0.00	99.96	1560.00
3.17	6.33	940.00	1.57	96.83	1260.00	0.01	99.96	1580.00
4.87	11.21	960.00	1.00	97.83	1280.00	0.01	99.98	1600.00
6.57	17.78	980.00	0.65	98.48	1300.00	0.00	99.98	1620.00
7.55	25.32	1000.00	0.40	98.88	1320.00	0.01	99.99	1640.00
8.26	33.58	1020.00	0.33	99.21	1340.00	0.00	99.99	1660.00
8.66	42.24	1040.00	0.24	99.45	1360.00	0.00	99.99	1680.00
8.85	51.09	1060.00	0.17	99.63	1380.00	0.00	99.99	1700.00
7.96	59.05	1080.00	0.08	99.71	1400.00	0.00	99.99	1720.00
7.07	66.12	1100.00	0.09	99.80	1420.00	0.01	100.00	1740.00
6.82	72.94	1120.00	0.07	99.87	1440.00	0.00	100.00	1749.00

Table 10. Cumulative Percent Frequency Distribution of Minimum Exospheric Temperature When 13-Month Smoothed 10.7-cm Solar Flux Is Between 138 and 174

PFD	CPF		PFD	CPF		PFD	CPF	
0.08	0.08	714.00	7.66	70.07	994.00	0.12	99.64	1274.00
0.31	0.39	734.00	6.70	76.77	1014.00	0.14	99.79	1294.00
1.14	1.53	754.00	6.06	82.84	1034.00	0.05	99.84	1314.00
2.07	3.60	774.00	4.77	87.60	1054.00	0.07	99.91	1334.00
2.24	5.84	794.00	3.79	91.40	1074.00	0.03	99.94	1354.00
3.02	8.87	814.00	2.47	93.87	1094.00	0.02	99.96	1374.00
4.16	13.03	834.00	1.80	95.66	1114.00	0.01	99.97	1394.00
5.32	18.35	854.00	1.39	97.06	1134.00	0.01	99.98	1414.00
5.94	24.29	874.00	0.83	97.88	1154.00	0.00	99.99	1434.00
6.58	30.87	894.00	0.55	98.43	1174.00	0.01	100.00	1454.00
8.15	39.02	914.00	0.40	98.82	1194.00	0.00	100.00	1474.00
8.26	47.28	934.00	0.29	99.12	1214.00	0.00	100.00	1494.00
7.69	54.97	954.00	0.19	99.31	1234.00			
7.44	62.41	974.00	0.21	99.52	1254.00			

Table 11. Cumulative Percent Frequency Distribution of Mean Exospheric Temperature When 13-Month Smoothed 10.7-cm Solar Flux Is Between 138 and 174

PFD	CPF		PFD	CPF		PFD	CPF	
0.08	0.08	836.00	7.00	69.56	1136.00	0.13	99.59	1436.00
0.31	0.39	856.00	6.51	76.07	1156.00	0.14	99.73	1456.00
0.95	1.34	876.00	5.63	81.71	1176.00	0.09	99.82	1476.00
1.67	3.01	896.00	4.70	86.41	1196.00	0.04	99.86	1496.00
2.20	5.21	916.00	3.89	90.30	1216.00	0.06	99.92	1516.00
2.54	7.74	936.00	2.59	92.90	1236.00	0.03	99.96	1536.00
3.35	11.09	956.00	1.91	94.80	1256.00	0.02	99.98	1556.00
4.50	15.60	976.00	1.46	96.27	1276.00	0.00	99.99	1576.00
5.38	20.98	996.00	1.02	97.29	1296.00	0.00	99.99	1596.00
5.61	26.58	1016.00	0.70	97.99	1316.00	0.01	100.00	1616.00
6.61	33.19	1036.00	0.52	98.51	1336.00	0.00	100.00	1636.00
7.93	41.12	1056.00	0.34	98.84	1356.00	0.00	100.00	1656.00
7.56	48.68	1076.00	0.28	99.12	1376.00	0.00	100.00	1659.00
7.03	55.71	1096.00	0.17	99.29	1396.00			
6.85	62.56	1116.00	0.17	99.46	1416.00			

Table 12. Cumulative Percent Frequency Distribution of Maximum Exospheric Temperature When 13-Month Smoothed 10.7-cm Solar Flux Is Between 138 and 174

PFD	CPF		PFD	CPF		PFD	CPF	
0.08	0.08	958.00	6.35	62.90	1258.00	0.21	99.30	1558.00
0.33	0.41	978.00	6.22	69.11	1278.00	0.14	99.43	1578.00
0.83	1.24	998.00	6.11	75.22	1298.00	0.15	99.58	1598.00
1.40	2.64	1018.00	5.47	80.69	1318.00	0.07	99.65	1618.00
2.04	4.68	1038.00	4.48	85.17	1338.00	0.11	99.76	1638.00
2.13	6.81	1058.00	3.76	88.93	1358.00	0.08	99.84	1658.00
2.98	9.79	1078.00	2.97	91.90	1378.00	0.05	99.90	1678.00
3.73	13.51	1098.00	1.95	93.85	1398.00	0.04	99.94	1698.00
4.54	18.05	1118.00	1.54	95.39	1418.00	0.03	99.97	1718.00
5.28	23.33	1138.00	1.21	96.60	1438.00	0.02	99.99	1738.00
5.43	28.75	1158.00	0.85	97.45	1458.00	0.00	99.99	1758.00
6.70	35.46	1178.00	0.61	98.06	1478.00	0.01	100.00	1778.00
7.38	42.84	1198.00	0.45	98.52	1498.00	0.00	100.00	1798.00
6.99	49.83	1218.00	0.33	98.84	1518.00	0.00	100.00	1818.00
6.71	56.34	1238.00	0.25	99.09	1538.00	0.00	100.00	1825.00

Table 13. Cumulative Percent Frequency Distribution of Minimum Exospheric Temperature When 13-Month Smoothed 10.7-cm Solar Flux Is Between 174 and 210

PFD	CPF		PFD	CPF		PFD	CPF	
0.02	0.02	764.00	6.99	55.04	1084.00	0.26	99.18	1404.00
0.11	0.13	784.00	6.60	61.64	1104.00	0.18	99.36	1424.00
0.12	0.25	804.00	5.87	67.51	1124.00	0.24	99.59	1444.00
0.29	0.54	824.00	5.81	73.32	1144.00	0.10	99.70	1464.00
0.72	1.27	844.00	4.99	78.32	1164.00	0.08	99.78	1484.00
1.53	2.79	864.00	4.16	82.48	1184.00	0.07	99.85	1504.00
2.79	5.58	884.00	3.99	86.47	1204.00	0.04	99.89	1524.00
3.25	8.83	904.00	3.48	89.94	1224.00	0.06	99.96	1544.00
3.45	12.28	924.00	2.65	92.90	1244.00	0.03	99.98	1564.00
3.12	15.39	944.00	1.94	94.54	1264.00	0.01	99.99	1584.00
4.23	19.62	964.00	1.45	95.99	1284.00	0.00	99.99	1604.00
4.78	24.40	984.00	1.19	97.18	1304.00	0.00	99.99	1624.00
5.34	29.74	1004.00	0.70	97.89	1324.00	0.00	99.99	1644.00
5.65	35.40	1024.00	0.52	98.41	1344.00	0.01	99.99	1664.00
6.17	41.56	1044.00	0.27	98.67	1364.00	0.01	100.00	1684.00
6.49	48.05	1064.00	0.24	98.92	1384.00	0.00	100.00	1696.00

Table 14. Cumulative Percent Frequency Distribution of Mean Exospheric Temperature When 13-Month Smoothed 10.7-cm Solar Flux Is Between 174 and 210

PFD	CPF		PFD	CPF		PFD	CPF	
0.03	0.03	894.00	6.29	52.11	1234.00	0.15	98.93	1574.00
0.10	0.13	914.00	6.23	58.35	1254.00	0.22	99.16	1594.00
0.10	0.23	934.00	5.75	64.10	1274.00	0.22	99.38	1614.00
0.19	0.42	954.00	5.40	69.50	1294.00	0.21	99.58	1634.00
0.41	0.83	974.00	4.91	74.41	1314.00	0.11	99.70	1654.00
1.08	1.91	994.00	4.38	78.79	1334.00	0.08	99.77	1674.00
1.99	3.90	1014.00	3.98	82.76	1354.00	0.04	99.82	1694.00
0.03	0.03	894.00	6.29	52.11	1234.00	0.15	98.93	1574.00
0.10	0.13	914.00	6.23	58.35	1254.00	0.22	99.16	1594.00
0.10	0.23	934.00	5.75	64.10	1274.00	0.22	99.38	1614.00
0.19	0.42	954.00	5.40	69.50	1294.00	0.21	99.58	1634.00
0.41	0.83	974.00	4.91	74.41	1314.00	0.11	99.70	1654.00
1.08	1.91	994.00	4.38	78.79	1334.00	0.08	99.77	1674.00
1.99	3.90	1014.00	3.98	82.76	1354.00	0.04	99.82	1694.00
2.98	6.88	1034.00	3.48	86.24	1374.00	0.06	99.88	1714.00
3.20	10.08	1054.00	3.33	89.57	1394.00	0.07	99.95	1734.00
3.06	13.14	1074.00	2.67	92.24	1414.00	0.02	99.97	1754.00
2.87	16.00	1094.00	1.77	94.02	1434.00	0.02	99.99	1774.00
3.53	19.53	1114.00	1.43	95.45	1454.00	0.00	99.99	1794.00
4.59	24.12	1134.00	1.19	96.64	1474.00	0.00	99.99	1814.00
5.06	29.19	1154.00	0.90	97.54	1494.00	0.01	99.99	1834.00
5.11	34.30	1174.00	0.64	98.18	1514.00	0.00	99.99	1854.00
5.84	40.14	1194.00	0.38	98.56	1534.00	0.01	100.00	1874.00
5.69	45.83	1214.00	0.23	98.79	1554.00	0.00	100.00	1876.00

Table 15. Cumulative Percent Frequency Distribution of Maximum Exospheric Temperature When 13-Month Smoothed 10.7-cm Solar Flux Is Between 174 and 210

PFD	CPF		PFD	CPF		PFD	CPF	
0.03	0.03	1024.00	5.63	49.93	1384.00	0.16	98.82	1744.00
0.10	0.12	1044.00	5.63	55.56	1404.00	0.13	98.95	1764.00
0.10	0.23	1064.00	5.45	61.01	1424.00	0.20	99.15	1784.00
0.14	0.36	1084.00	5.34	66.35	1444.00	0.21	99.36	1804.00
0.25	0.61	1104.00	4.76	71.11	1464.00	0.19	99.55	1824.00
0.71	1.33	1124.00	4.26	75.37	1484.00	0.15	99.70	1844.00
1.44	2.77	1144.00	3.81	79.18	1504.00	0.06	99.76	1864.00
2.43	5.20	1164.00	3.67	82.85	1524.00	0.05	99.81	1884.00
2.91	8.11	1184.00	3.27	86.13	1544.00	0.05	99.85	1904.00
3.05	11.16	1204.00	3.03	89.16	1564.00	0.06	99.91	1924.00
2.73	13.89	1224.00	2.55	91.71	1584.00	0.04	99.95	1944.00
2.73	16.62	1244.00	1.93	93.64	1604.00	0.03	99.98	1964.00
3.09	19.71	1264.00	1.34	94.98	1624.00	0.01	99.99	1984.00
4.34	24.05	1284.00	1.18	96.16	1644.00	0.01	99.99	2004.00
4.66	28.72	1304.00	0.95	97.11	1664.00	0.00	99.99	2024.00
5.03	33.75	1324.00	0.78	97.89	1684.00	0.01	100.00	2044.00
5.15	38.90	1344.00	0.49	98.37	1704.00	0.00	100.00	2055.00
5.40	44.30	1364.00	0.28	98.65	1724.00			

Table 16. Cumulative Percent Frequency Distribution of Minimum Exospheric Temperature When 13-Month Smoothed 10.7-cm Solar Flux Is Between 210 and 246

PFD	CPF		PFD	CPF		PFD	CPF	
0.03	0.03	924.00	5.50	61.90	1224.00	0.50	98.89	1524.00
0.34	0.37	944.00	5.07	66.97	1244.00	0.34	99.23	1544.00
0.69	1.06	964.00	4.41	71.39	1264.00	0.27	99.50	1564.00
1.26	2.31	984.00	3.89	75.27	1284.00	0.16	99.66	1584.00
1.30	3.61	1004.00	3.44	78.71	1304.00	0.11	99.77	1604.00
1.94	5.56	1024.00	3.67	82.89	1324.00	0.04	99.81	1624.00
3.43	8.99	1044.00	2.80	85.19	1344.00	0.07	99.89	1644.00
4.93	13.91	1064.00	2.20	87.39	1364.00	0.09	99.97	1664.00
5.26	19.17	1084.00	2.37	89.76	1384.00	0.01	99.99	1684.00
5.44	24.61	1104.00	1.91	91.67	1404.00	0.00	99.99	1704.00
5.50	30.11	1024.00	2.17	93.84	1424.00	0.00	99.99	1724.00
6.21	36.33	1144.00	1.59	95.43	1444.00	0.00	99.99	1744.00
6.84	43.17	1164.00	1.27	96.70	1464.00	0.01	100.00	1764.00
6.83	50.00	1184.00	0.86	97.56	1484.00	0.00	100.00	1767.00
6.40	56.40	1204.00	0.83	98.39	1504.00			

Table 17. Cumulative Percent Frequency Distribution of Mean Exospheric Temperature When 13-Month Smoothed 10.7-cm Solar Flux Is Between 210 and 246

PFD	CPF		PFD	CPF		PFD	CPF	
0.03	0.03	1084.00	4.06	62.06	1404.00	0.47	98.74	1724.00
0.30	0.33	1104.00	4.49	66.54	1424.00	0.41	99.16	1744.00
0.57	0.90	1124.00	4.24	70.79	1444.00	0.21	99.37	1764.00
1.21	2.11	1144.00	3.69	74.47	1464.00	0.20	99.57	1784.00
1.06	3.17	1164.00	3.34	77.81	1484.00	0.17	99.74	1804.00
1.60	4.77	1184.00	3.20	81.01	1504.00	0.06	99.80	1824.00
2.50	7.27	1204.00	2.86	83.87	1524.00	0.06	99.86	1844.00
4.53	11.80	1224.00	2.03	85.90	1544.00	0.10	99.96	1864.00
5.49	17.29	1244.00	2.01	87.91	1564.00	0.01	99.97	1884.00
4.74	22.03	1264.00	2.24	90.16	1584.00	0.01	99.99	1904.00
5.56	27.59	1284.00	1.97	92.13	1604.00	0.00	99.99	1924.00
5.40	32.99	1304.00	1.81	93.94	1624.00	0.00	99.99	1944.00
6.11	39.10	1324.00	1.64	95.59	1644.00	0.01	100.00	1964.00
6.56	45.66	1344.00	0.96	96.54	1664.00	0.00	100.00	1975.00
6.34	52.00	1364.00	0.90	97.44	1684.00			
6.00	58.00	1384.00	0.83	98.27	1704.00			

Table 18. Cumulative Percent Frequency Distribution of Maximum Exospheric Temperature When 13-Month Smoothed 10.7-cm Solar Flux Is Between 210 and 246

PFD	CPF		PFD	CPF		PFD	CPF	
0.03	0.03	1243.00	3.89	62.19	1583.00	0.64	98.57	1923.00
0.21	0.24	1263.00	3.64	65.83	1603.00	0.40	98.97	1943.00
0.56	0.80	1283.00	4.06	69.89	1623.00	0.36	99.33	1963.00
0.97	1.77	1303.00	3.60	73.49	1643.00	0.09	99.41	1983.00
1.11	2.89	1323.00	3.31	76.80	1663.00	0.19	99.60	2003.00
1.39	4.27	1343.00	3.01	79.81	1683.00	0.14	99.74	2023.00
1.94	6.21	1363.00	2.49	82.30	1703.00	0.13	99.87	2043.00
3.39	9.60	1383.00	2.36	84.66	1723.00	0.06	99.93	2063.00
5.44	15.04	1403.00	1.97	86.63	1743.00	0.03	99.96	2083.00
4.86	19.90	1423.00	1.71	88.34	1763.00	0.03	99.99	2103.00
5.19	25.09	1443.00	2.19	90.53	1783.00	0.00	99.99	2123.00
4.67	29.76	1463.00	1.89	92.41	1803.00	0.00	99.99	2143.00
5.91	35.67	1483.00	1.41	93.83	1823.00	0.00	99.99	2163.00
5.69	41.36	1503.00	1.70	95.53	1843.00	0.01	100.00	2183.00
6.14	47.50	1523.00	0.94	96.47	1863.00			
5.89	53.39	1543.00	0.80	97.27	1883.00			
4.91	58.30	1563.00	0.66	97.93	1903.00			

PROGRAM LISTINGS

SUBROUTINE J70MET (TE, Z, TZ, DENS, DL)

```
C..J70MET developed from MET2 by Mike Hickey, Physitron Consultant

C..INPUTS:
C.. TE  -- exospheric temperature
C.. Z   -- altitude
C..
C..OUTPUTS:
C.. TZ  -- temperature at altitude Z
C.. DENS -- total density at altitude Z
C.. DL  -- log10 total density at altitude Z

      PARAMETER RGAS = 8.31432E3          ! J/kmol-K
      PARAMETER BFH = 440.0

C Calcultions performed for only one latitude, one longitude
C and one altitude
      CALL JAC (Z, TE, TZ, EM, DENS, DL)

      RETURN
      END
```

SUBROUTINE JAC (Z, T, TZ, EM, DENS, DL)

```
C  Subroutine 'JAC' calculates the temperature TZ, the total density DENS
C  and its logarithm DL, the mean molecular weight EM, the individual
C  specie number densities for N, O2, O, A, HE and H (each preceded with
C  an 'A') at altitude Z given the exospheric temperature T.
C  This subroutine uses the subroutine 'GAUSS' and the function
C  subprograms 'TEMP' and 'MOL_WT'.
C
C  Originally Rewritten by Mike Hickey, USRA


        DIMENSION  ALPHA(6), EI(6), DI(6), DIT(6)
        REAL*4  MOL_WT

        PARAMETER AV   = 6.02257E23
        PARAMETER QN   = .78110
        PARAMETER QO2  = .20955
        PARAMETER QA   = .009343
        PARAMETER QHE  = 1.289E-5
        PARAMETER RGAS = 8.31432
        PARAMETER PI   = 3.14159265
        PARAMETER T0   = 183.

        GRAVITY (ALTITUDE) = 9.80665 / (( 1. + ALTITUDE / 6.356766E3)**2)

        DATA ALPHA / 0.0, 0.0, 0.0, 0.0, -.380, 0.0 /
        DATA EI / 28.0134, 31.9988, 15.9994, 39.948, 4.0026, 1.00797 /

        TX = 444.3807 + .02385 * T - 392.8292 * EXP (-.0021357 * T)
        A2 = 2. * (T-TX) / PI
        TX_T0 = TX - T0
        T1 =  1.9 * TX_T0 /  35.
        T3 = -1.7 * TX_T0 / (35.**3)
        T4 = -0.8 * TX_T0 / (35.**4)
        TZ = TEMP (Z, TX, T1, T3, T4, A2)

C       SECTION 1
C       ---------

        A = 90.
        D = AMIN1 (Z, 105.)
```

```
C Integrate gM/T from 90 to minimum of Z or 105 km :-

      CALL GAUSS (A, D, 1, R, TX, T1, T3, T4, A2)

C The number 2.1926E-8 = density x temperature/mean molecular weight at 90 km.
      EM = MOL_WT (D)
      TD = TEMP (D, TX, T1, T3, T4, A2)

      DENS = 2.1926E-8*EM*EXP(-R/RGAS)/TD
      FACTOR = AV*DENS
      PAR = FACTOR / EM
      FACTOR = FACTOR / 28.96

C For altitudes below and at 105 km calculate the individual specie number
C densities from the mean molecular weight and total density.
      IF (Z. LE. 105) THEN
         DL = ALOG10 (DENS)
         AN  = ALOG10 (QN*FACTOR)
         AA  = ALOG10 (QA*FACTOR)
         AHE = ALOG10 (QHE*FACTOR)
         AO  = ALOG10 (2.*PAR*(1.-EM / 28.96))
         AO2 = ALOG10 (PAR*(EM*(1.+QO2) / 28.96-1.))
         AH  = 0.
C Return to calling program
         RETURN
      END IF


C      SECTION 2 :  This section is only performed for altitudes above 105 km
C      --------
C Note that having reached this section means that D in section 1 is 105 km.
C Calculate individual specie number densities from the total density and mean
C molecular weight at 105 km altitude.
      DI(1) = QN*FACTOR
      DI(2) = PAR*(EM*(1.+QO2) / 28.96-1.)
      DI(3) = 2.*PAR*(1.- EM / 28.96)
      DI(4) = QA*FACTOR
      DI(5) = QHE*FACTOR

C Integrate g/T from 105 km to Z km :-
      CALL GAUSS (D, Z, 2, R, TX, T1, T3, T4, A2)

      DO 41 I = 1, 5
         DIT(I) = DI(I)*(TD/TZ)**(1.+ALPHA(I))*EXP(-EI(I)*R/RGAS)
```

```fortran
      IF (DIT(I). LE. 0.) DIT(I) = 1.E-6
41    CONTINUE

C  This section calculates atomic hydrogen densities above 500 km altitude.
C  Below this altitude, H densities are set to 10-6.
C     SECTION 3
C     ---------

      IF (Z .GT. 500.)  THEN
        A1 = 500.
        S = TEMP (A1, TX, T1, T3, T4, A2)
        DI(6) = 10.**(73.13 - 39.4*ALOG10 (S) + 5.5*ALOG10(S)*ALOG10(S))
        CALL GAUSS (A1, Z, 7, R, TX, T1, T3, T4, A2)
        DIT(6) = DI(6)*(S/TZ)*EXP (-EI(6)*R / RGAS)
      ELSE
        DIT (6) = 1.0
      END IF

C  For altitudes greater than 105 km, calculate total density and mean
C  molecular weight from individual specie number densities.

      DENS=0
      DO  42  I = 1, 6
        DENS = DENS + EI(I)*DIT(I) / AV
42    CONTINUE

      EM = DENS*AV / (DIT(1)+DIT(2)+DIT(3)+DIT(4)+DIT(5)+DIT(6))
      DL = ALOG10 (DENS)

      AN  = ALOG10(DIT(1))
      AO2 = ALOG10(DIT(2))
      AO  = ALOG10(DIT(3))
      AA  = ALOG10(DIT(4))
      AHE = ALOG10(DIT(5))
      AH  = ALOG10(DIT(6))

      RETURN
      END
```

```fortran
      FUNCTION  TEMP (ALT, TX, T1, T3, T4, A2)
C Function subprogram 'TEMP' calculates the temperature at altitude ALT
C using equation (10) for altitudes between 90 and 125 km and equation
C (13) for altitudes greater than 125 km, from SAO Report 313.
C
C Written by Mike Hickey, USRA

      PARAMETER BB = 4.5E-6

      U = ALT - 125.
      IF (U .GT . 0.)  THEN
        TEMP = TX  +  A2*ATAN (T1*U*(1. + BB*(U**2.5)) / A2)
      ELSE
        TEMP = TX  +  T1*U  +  T3*(U3**3)  +  T4*(U4**4)
      END IF

      END




      REAL FUNCTION MOL_WT*4 (A)
C Subroutine 'MOL_WT' calculates the molecular weight for altitudes
C between 90 and 105 km according to equation (1) of SAO report 313.
C Otherwise, MOL_WT is set to unity.
C
C Written by Mike Hickey, USRA

      DIMENSION  B (7)

      DATA B / 28.15204, -0.085586, 1.284E-4, -1.0056E-5, -1.021E-5,
               1.5044E-6, 9.9826E-8 /

      IF (A. GT. 105.) THEN
        MOL_WT = 1.
      ELSE
        U = A - 100.
        MOL_WT = B (1)
        DO 1 I = 2, 7
          MOL_WT = MOL_WT  +  B (I)*U**(I-1)
1       CONTINUE
      END IF

      END
```

```
      SUBROUTINE GAUSS (Z1, Z2, NMIN, R, TX, T1, T3, T4, A2)
C Subdivide total integration-altitude range into intervals suitable for
C applying Gaussian Quadrature, set the number of points for integration
C for each sub-interval, and then perform Gaussian Quadrature.
C Written by Mike Hickey, USRA, NASA/MSFC, ED44, July 1988.

      REAL*4  ALTMIN (9), C(8,6), X(8,6), MOL_WT
      INTEGER  NG (8), NGAUSS, NMIN, J

      GRAVITY (ALTITUDE) = 9.80665 / (( 1. + ALTITUDE / 6.356766E3)**2)

      DATA ALTMIN / 90., 105., 125., 160., 200., 300., 500., 1500., 2500. /
      DATA   NG  / 4, 5 , 6 , 6 , 6 , 6 , 6 , 6 /


C Coefficients for Gaussian Quadrature ...

      DATA C / .5555556 , .8888889 , .5555556 , .0000000 , ! n=3
   .         .0000000 , .0000000 , .0000000 , .0000000 , ! n=3
   .         .3478548 , .6521452 , .6521452 , .3478548 , ! n=4
   .         .0000000 , .0000000 , .0000000 , .0000000 , ! n=4
   .         .2369269 , .4786287 , .5688889 , .4786287 , ! n=5
   .         .2369269 , .0000000 , .0000000 , .0000000 , ! n=5
   .         .1713245 , .3607616 , .4679139 , .4679139 , ! n=6
   .         .3607616 , .1713245 , .0000000 , .0000000 , ! n=6
   .         .1294850 , .2797054 , .3818301 , .4179592 , ! n=7
   .         .3818301 , .2797054 , .1294850 , .0000000 , ! n=7
   .         .1012285 , .2223810 , .3137067 , .3626838 , ! n=8
   .         .3626838 , .3137067 , .2223810 , .1012285 / ! n=8


C Abscissas for Gaussian Quadrature ...

      DATA X / -.7745967 , .0000000 , .7745967 , .0000000 , ! n=3
   .         .0000000 , .0000000 , .0000000 , .0000000 , ! n=3
   .        -.8611363 , -.3399810 , .3399810 , .8611363 , ! n=4
   .         .0000000 , .0000000 , .0000000 , .0000000 , ! n=4
   .        -.9061798 , -.5384693 , .0000000 , .5384693 , ! n=5
   .         .9061798 , .0000000 , .0000000 , .0000000 , ! n=5
   .        -.9324695 , -.6612094 , -.2386192 , .2386192 , ! n=6
   .         .6612094 , .9324695 , .0000000 , .0000000 , ! n=6
   .        -.9491079 , -.7415312 , -.4058452 , .0000000 , ! n=7
   .         .4058452 , .7415312 , .9491079 , .0000000 , ! n=7
   .        -.9602899 , -.7966665 , -.5255324 , -.1834346 , ! n=8
   .         .1834346 , .5255324 , .7966665 , .9602899 / ! n=8
```

```fortran
      R = 0.0

      DO  2  K = NMIN, 8
        NGAUSS = NG (K)
A     = ALTMIN (K)
D     = AMIN1 (Z2, ALTMIN (K+1))
RR    = 0.0
DEL = 0.5*(D - A)
J   = NGAUSS - 2
        DO  1  I = 1, NGAUSS
          Z = DEL*(X(I,J) + 1.) + A
          RR = RR + C(I,J)*MOL_WT(Z)*GRAVITY(Z) / TEMP (Z,TX,T1,T3,T4,A2)
1       CONTINUE
RR = DEL*RR
R  = R + RR
IF (D .EQ. Z2) RETURN
2       CONTINUE

      RETURN
      END
```

```
c...test routine for j70met
      IMPLICIT    NONE
      REAL        TE, Z, TZ, DENS, DL

      TYPE *,'Input exospheric temperature and altitude (km)'
      ACCEPT *, TE, Z

      CALL J70MET (TE, Z, TZ, DENS, DL)
      TYPE *,' T = ',TZ
      TYPE *,' Density = ',DENS
      TYPE *,' log(density) = ',DL
      STOP
      END
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE May 1998	3. REPORT TYPE AND DATES COVERED Contractor Report (Final)

4. TITLE AND SUBTITLE The Marshall Engineering Thermosphere (MET) Model Volume I: Technical Description	5. FUNDING NUMBERS NAS8–38333
6. AUTHORS R.E. Smith	

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES) Physitron, Inc. 3304 Westmill Drive Huntsville, AL 35805	8. PERFORMING ORGANIZATION REPORT NUMBER M–866

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) George C. Marshall Space Flight Center Marshall Space Flight Center, AL 35812	10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA/CR–1998–207946

11. SUPPLEMENTARY NOTES

Prepared for Systems Analysis and Integration Laboratory, Science and Engineering Directorate
Technical Monitor: Dr. B. Jeffrey Anderson

12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified–Unlimited Subject Category 18 Standard Distribution	12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Volume I presents a technical description of the Marshall Engineering Thermosphere (MET) model atmosphere and a summary of its historical development. Various programs developed to augment the original capability of the model are discussed in detail. The report also describes each of the individual subroutines developed to enhance the model. Computer codes for these subroutines are contained in four appendices.

Volume II will not be released as a formal report. This volume contains a compilation of references describing the various components of the total model program and details of the development of the various options that can be exercised in the model. Volume II will be made available to qualified requesters who contact the Chief, Electromagnetics and Aerospace Environments Branch, Systems Analysis and Integration Laboratory, MSFC, Alabama 35812.

14. SUBJECT TERMS atmospheric models, thermosphere, solar activity indices, thermospheric density, gravity waves	15. NUMBER OF PAGES 105
	16. PRICE CODE A06

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited