NASA CONTRACTOR
REPORT

NASA CR-179359

THE NASA MARSHALL ENGINEERING THERMOSPHERE MODEL

(NASA-CR-179359) THE NASA MARSHALL ENGINEERING THERMOSPHERE MODEL (Universities Space Research Association) 40 p CSCL 04A

N88-26760

Unclas
G3/46 0156149

By M. P. Hickey
Universities Space Research Association
4950 Corporate Drive, Suite 100
Huntsville, Alabama 35806

July 1988

Interim Report

Prepared for
NASA, George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
NASA CR-179359		

4. TITLE AND SUBTITLE	5. REPORT DATE
The NASA Marshall Engineering Thermosphere Model	July 1988
	6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) Michael Philip Hickey	8. PERFORMING ORGANIZATION REPORT #
9. PERFORMING ORGANIZATION NAME AND ADDRESS Universities Space Research Association 4950 Corporate Drive, Suite 100 Huntsville, Alabama 35806	10. WORK UNIT NO. 11. CONTRACT OR GRANT NO. NAS8-36400
12. SPONSORING AGENCY NAME AND ADDRESS National Aeronautics and Space Administration Washington, DC 20546	13. TYPE OF REPORT & PERIOD COVERED Contractor Report 14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

Technical Monitor: Claude E. Green, Structures and Dynamics Lab, Science and Engineering Directorate, George C. Marshall Space Flight Center, MSFC, Alabama

16. ABSTRACT

This document contains a description of the NASA Marshall Engineering Thermosphere (MET) Model, which is a modified version of the MSFC/J70 Orbital Atmospheric Density Model as currently used in the J70MM program at MSFC. The modifications to the MSFC/J70 model required for the MET model are described, graphical and numerical examples of the models are included, as is a listing of the MET model computer program. Major differences between the numerical output from the MET model and the MSFC/J70 model are discussed.

17. KEY WORDS	18. DISTRIBUTION STATEMENT
Density Thermosphere Model Computer-Program	Unclassified/Unlimited

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	39	NTIS

MSFC - Form 3292 (Rev. December 1972)

For sale by National Technical Information Service, Springfield, Virginia 22151

PREFACE

This work was performed under NASA Contract NAS8-36400 under the technical monitorship of Messrs. Claude E. Green and Dale L. Johnson. Qualified requestors may obtain copies of the computer program for this NASA Marshall Engineering Thermosphere model upon request to the Chief, Earth Science and Applications Division, Structures and Dynamics Laboratory, NASA, Marshall Space Flight Center, Alabama 35812.

TABLE OF CONTENTS

CONTRACTOR REPORT

THE NASA MARSHALL ENGINEERING THERMOSPHERE MODEL

INTRODUCTION

The model currently used at NASA/Marshall Space Flight Center to describe the properties of the neutral atmosphere between 90 and 2500 km altitude is the MSFC/J70 model [1,2], as described in [3]. In the Earth Science and Applications Division of the Structures and Dynamics Laboratory the computer program used to output data from the MSFC/J70 model is the J70MM. This program, which runs on an HP-1000 computer, outputs data in a latitude/longitude matrix grid (for a fixed time, altitude and solar and geomagnetic indices) which can then be printed on a line printer. A listing of this program, which is written in Fortran IV, can be found in [3].

For many purposes it is more convenient to have any computer generated output plotted rather than simply tabulated, and with several different input parameters used in the model (solar and geomagnetic indices, latitude, longitude, altitude and time) it was decided that the computer code should be substantially modified and streamlined. This was implemented on a VAX 11/780 computer by the use of more subroutines and by writing the code in Fortran 77. Some other modifications were also performed which relate to the calculation of ephemeris data and to the calculation of helium number densities in the upper thermosphere. Some additional atmospheric thermodynamic parameters are now also calculated in the model.

In the course of producing global contour plots from the data produced by the new streamlined model, it was found that occasionally density discontinuities were present over certain regions of the globe. On further investigation, it was discovered that these discontinuities were associated with an occasional problem in the integration routine used in the model. More details concerning the nature of this problem and its solution are given in this report.

The final version of this model has been named the NASA Marshall Engineering Thermosphere (MET) Model. It consists of a number of subroutines which are driven by a very simple driving program, as given in the appendix. To produce plots, all one needs to do is modify the driving program. This has been done in an accompanying report [4], the details of which will not appear here.

SUBROUTINE STRUCTURING

A schematic representation of the MET model is given in Figure 1. The driving program calls the control subroutine ATMOSPHERES, which then makes decisions based upon the values of the four switches which have been set in the driver. In the basic model presented here, only one calculation is performed (ie. calculations are performed for one altitude, latitude, longitude and time and for one set of solar and geomagnetic conditions) so that all of the switches have been set to a character value of "Y". The development of a driving program suitable for the generation of data files to be used for creating plots would require the execution of multiple calculations, and hence would require one of the switches being set to a character value of "N" in the driver program for an x-y line plot, or would require at least one or two of the switches being set to a character value of "N" in the driver program for a contour plot. The subject of multiple executions is not discussed any further here, but is discussed in [4].

Once the subroutine ATMOSPHERES has been called, it successively calls the subroutines TIME, SOLSET, J70 and J70SUP. Here, in this most basic setup, TIME and SOLSET are called once only. Each of these subroutines prompt the user for the time (year, month, day, hour and minute) and the solar and geomagnetic indices (AP, F10.7 and F10.7), respectively, as discussed in [3]. In the MET model the user, not given a choice of using the KP index, could easily edit SOLSET should the KP index be preferred over the use of the AP index.

Subroutine J70 calculates most of the atmospheric parameters which the MSFC/J70 model did, but it contains several modifications which will be discussed shortly. J70 successively calls the subroutines TME, TINF, JAC, SLV and SLVH. Apart from changing the code from Fortran IV to Fortran 77, TINF, SLV and SLVH are basically very similar to their old versions appearing in the MSFC/J70 model. Some constants have been changed in TME, while JAC has been extensively re-written in an attempt to make it intelligible to the reader! An intermittent problem within JAC has also been corrected, and this will be discussed in more detail shortly. Between altitudes of 440 and 500 km J70 also calls a new subroutine called FAIR5, which we will discuss shortly.

Once calculations in J70 are finished, J70SUP is called. This subroutine calculates additional atmospheric thermodynamic parameters which will also be described shortly.

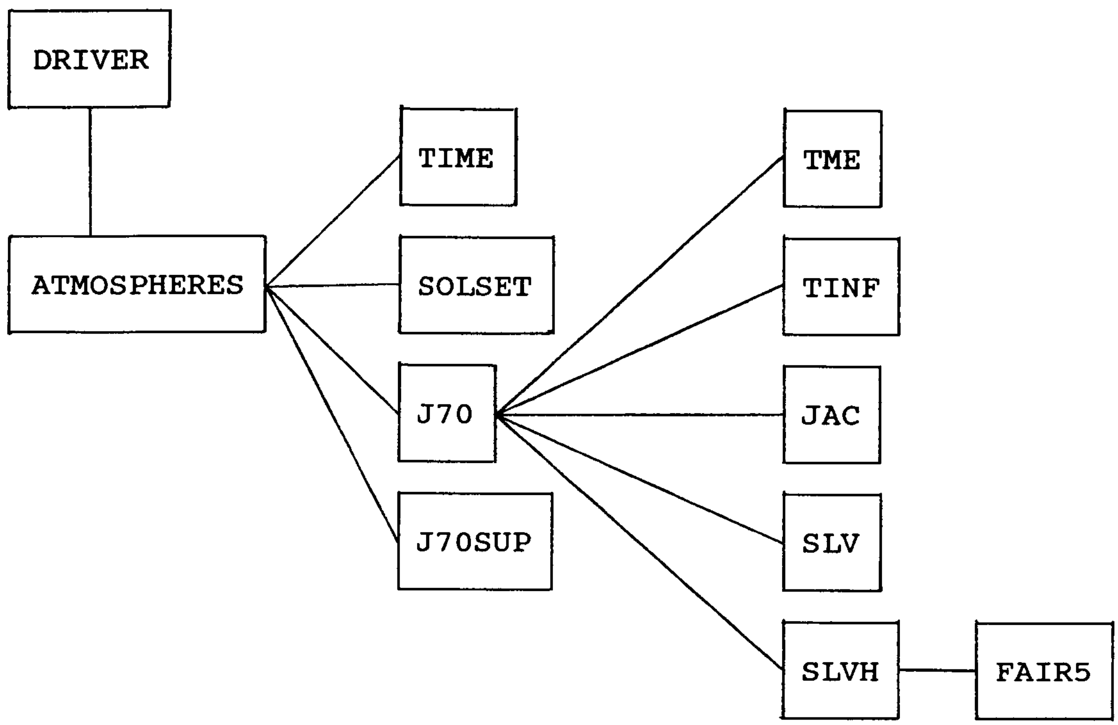

Figure 1. The MET MODEL*

*Sequential numerical operation is down and to the right

SUBROUTINE TME

A number of relatively minor changes have been made to some of the ephemeris constants, as described in [5]. These changes, listed in Table 1, lead to improvements in the calculated values of the solar declination and solar hour angles, as given in [5]. However, the associated changes in the total density are typically fractions of a percent.

TABLE 1

PARAMETER	OLD VALUE	NEW VALUE
A2	36000.76854	36000.76892
A4	0.25068447	0.250684477
B1	0.017203	0.0172028
B3	1.410	1.407
FMJD	XMJD-2435839	XMJD-2435839+GMT/1440.
XJ	(XMJD-2415020.0)/36525.0	(XMJD-2415020.5)/36525.0
XLS	AMOD(Y1+B2*SIN(YI)-B3,TPI)	AMOD(YI+B2*SIN(Y2)-B3,TPI) where Y2=0.017202*(FMJD-3.)
B4	(TPI/360.)*23.45	(TPI/360.)*(23.4523-0.013*XJ)

SUBROUTINE JAC

This subroutine has been extensively streamlined and improved by the introduction of a function statement (GRAVITY), two function subprograms (TEMP and MOL_WT) and a subroutine (INTEGRATE). Section 1 of JAC is used for altitudes between 90 and 105 km, while Section 2 is used only for altitudes greater than 105 km. Section 3 is used only for the calculation of hydrogen number densities which below 500 km altitude are set equal to $10^6 m^{-3}$.

The function subprogram MOL_WT only calculates the molecular weight at and below altitudes of 105 km using the sixth-degree polynomial given in [1]. Given this molecular weight and the temperature, the total mass density is computed by integrating the barometric equation. For altitudes above 105 km the number density of each individual specie is calculated by integrating the diffusion equation upwards from 105 km. Thus, above 105 km altitude the molecular weight is not known (it is calculated later from the number densities of all of the species) nor is it needed. Thus, the subprogram MOL_WT simply sets the molecular weight to unity above 105 km altitude.

SUBROUTINE INTEGRATE

Between 90 and 105 km altitude the atmosphere is well mixed, but as a result of the dissociation of O_2 to O the mean molecular mass in the model varies in such a manner that the ratio $n(o)/n(O_2)=1.5$ at 120 km [1]. Above 105 km each individual specie is in a state of diffusive equilibrium. The only exception to this is hydrogen, which in the model exists above 500 km in a state of diffusive equilibrium. Thus, between 90 and 105 km the density is computed by integrating the barometric equation while above 105 km it is computed by integrating the diffusion equation. This integration is performed in the subroutine INTEGRATE using Simpson's Rule.

Integration is performed between the altitudes A and D where at these altitudes the integrand has values of FA and FD, respectively. At and below 105 km altitude gM/T is integrated (g is the gravitational acceleration, M is the mean molecular weight and T is temperature), while above 105 km altitude g/T is integrated. Integration is performed using Simpson's Rule quadrature, and the value of the integral of either gM/T or g/T between altitudes A and D is set equal to R(N), where the number of subintervals used to evaluate the integral in the interval A to D is given by 2^{N-1}. Integration is said to have converged when the solution obtained by doubling the number of subintervals is "equal" to the previous solution within some specified tolerance, ϵ. Mathematically, this is stated as:

$$\epsilon |R(N)| > |R(N)-R(N-1)| \tag{1}$$

If equation (1) is satisfied, then convergence is said to have been achieved, the result (RR) is set to R(N), and execution returns to the calling program (in this case subroutine JAC). This method is the same as that used in the original J70MM program, but the code has been substantially re-written.

While there is nothing wrong with this basic method, the implementation of it in J70MM contained a major flaw which appears to be intermittent in nature. Figure 2(a) is a contour plot of total mass density over a portion of the globe at 230 km altitude. The input conditions required to generate this plot are:-

Time:	21 June, 1987, 0400 hrs.
Indices:	$F_{10.7} = 230$, $\bar{F}_{10.7} = 230$, $A_p = 0$
Altitude:	230 km

This plot was not generated with the MET model, but with a modified form of the original J70MM program. The important feature of this plot is not the actual magnitudes of the densities, but the discontinuous nature of these densities as a function of both latitude and longitude, as revealed by the several "saw-toothed curves" which are clearly evident in the plot. It is important to note, however, that all of the empirical equations upon which the model is based are continuous functions of both latitude and longitude [1]. Also, each individual data point used in Figure 2(a) is independent of all of the other data points which are used. Hence, any error in the model which has produced the discontinuities in this figure are peculiar to the calculations made for each individual data point which is in error. Due to the fact that a contour plot of mass density at 200 km altitude, (with all other input parameters identical to those used in Figure 2(a)) appeared to be perfectly continuous over latitude and longitude, a discontinuity in density as a function of altitude was sought.

Figure 2(b) shows mass density as a function of altitude for the same time and solar and geomagnetic indices as previously used in Figure 2(a), and for a latitude of -62.1^{o} and a longitude of 295^{o}. [This geographic position was chosen because it corresponds to a region of discontinuity in Figure 2(a). Also, only a small altitude range centered about 230 km was considered. Had a large altitude range been chosen, any small discontinuities may have been hidden on the plot by the scaling required to display the rapid exponential decrease of density which occurs over increasing altitude.] Immediately evident in this figure is a density discontinuity centered at approximately 229.8 km altitude with a vertical extent of approximately 1 km and a magnitude of about 1.2%. This discontinuity is a direct result of the convergence criterion used in the integration procedure, and although a detailed description of this problem is beyond the scope of this report, a brief description of the problem and its solution follows.

As already stated, integration is performed using Simpson's Rule quadrature. The form of this procedure, as used here, is an iterative one, and is such that unnecessary re-evaluation of the integrand is not required at the "old" subinterval end- and mid-points when the number of subintervals is doubled. Specifically, if the number of subintervals is 2^J, then the area A(J) is given by

$$A(J) = 1/3\ (SONE)_J + 4/3h_J\ (STOW)_J \qquad (2)$$

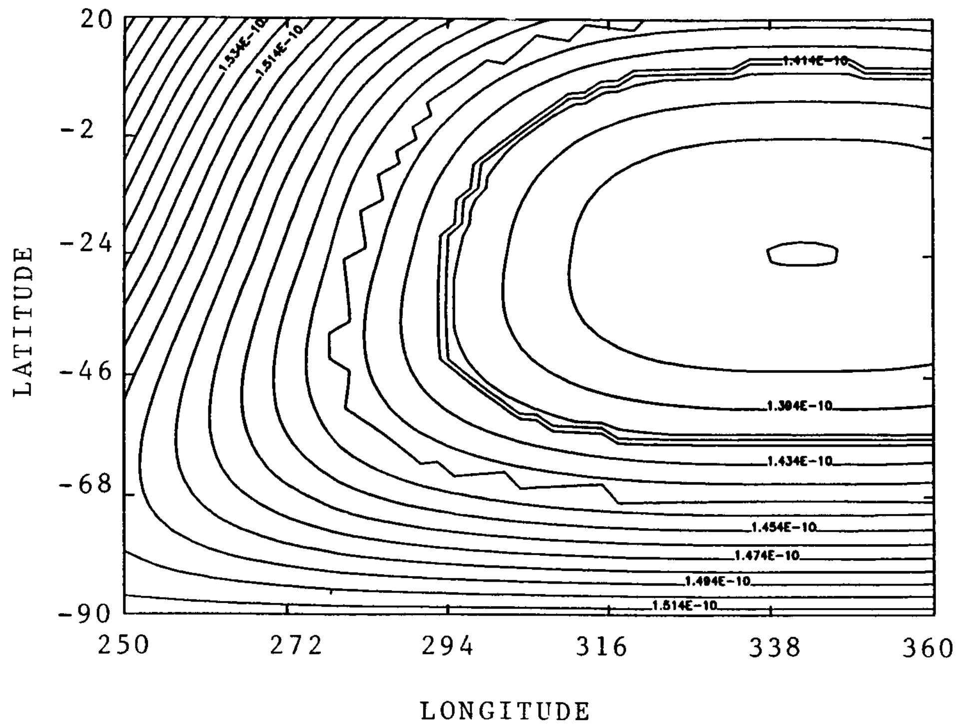

Figure 2(a) Contour plot of J70MM model mass density

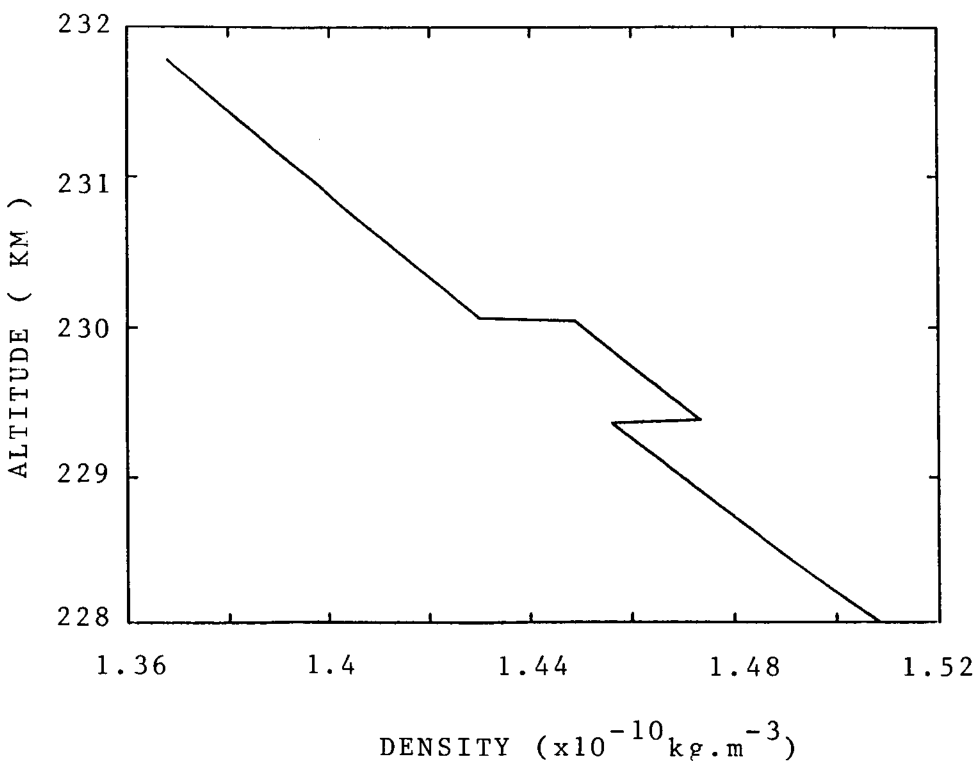

Figure 2(b) Altitude plot of J70MM model mass density where h_J is the subinterval length (equal to $(D-A).2^{-J}$ in subroutine INTEGRATE) and where SONE and STOW are functions of the integrand evaluated at subinterval end- and mid-points (the exact functional forms of SONE and STOW are not required here, but can be found in subroutine INTEGRATE). If the number of subintervals is now doubled (ie. 2^{J+1}), then the area calculated with this new number of subintervals (A(J+1)) in terms of A(J) is

$$A(J+1) = 1/2\ A(J) - 2/3\ h_{J+1}\ (STOW)_J + 4/3\ h_{J+1}\ (STOW)_{J+1} \quad (3)$$

Evaluation of A(J+1) simply requires the evaluation of $(STOW)_{J+1}$, because A(J) and $(STOW)_J$ are already known. Evaluation of $(STOW)_{J+1}$ requires calculating the integrand at every point lying halfway between every "old" subinterval (2^J of them) end- and mid-point.

Examination of equation (3) reveals that if

$$4/3\ h_{J+1}\ (STOW)_{J+1} - 2/3\ h_{J+1}\ (STOW)_J \approx 1/2\ A(J) \quad (4)$$

then iterative convergence, as defined in equation (1), may be achieved if the equality in equation (4) is "sufficiently close" (this will, of course, depend on the value of ϵ). That equation (4), and hence that equation (1), could be satisfied by fluke is not beyond the realm of possibility, and in fact as it transpires, the discontinuities in Figures 2(a) and 2(b) are directly attributable to this fluke convergence. For altitudes below about 229.3 km and above about 230.3 km the integration procedure required that at least 2^5 (ie. 32) subintervals be used to ensure convergence. Between these two altitudes only 2^3 (ie. 8) subintervals were required for equation (1) to be satisfied, meaning that no significant increase in accuracy was achieved by increasing the number of subintervals from 4 to 8. However, a re-calculation using 16 subintervals showed that convergence had not been achieved. In fact, 32 subintervals were required to ensure convergence.

The remedy to this problem, which was implemented in, and subsequently incorporated into INTEGRATE, is a very simple one but may not be guaranteed under all conditions. Quite simply, above 180 km altitude, the number of subintervals used must be at least 16 (ie. 2^4) before convergence is tested using equation (1). Up until the present time this has always been successful, but any future encounters with problems related to this should be reported to either the author or to the Branch Chief, ED44, NASA/MSFC, Huntsville, Alabama 35812.

The results of using the new MET model, which incorporates these modifications, are shown in figures 3(a) and 3(b). The input parameters used to generate these results are identical to those used for the corresponding results (using a modified J70MM program) shown in Figure 2. The MET model shows no sign of any discontinuities.

SUBROUTINE FAIR5

In the MET model, as in the J70MM model, the helium number density (and concomitantly the total mass density) above 500 km altitude is modified by seasonal-latitudinal variations according to the equations of Jacchia [2]. This seasonal-latitudinal variation of helium is calculated in subroutine SLVH. (This should not be confused with the seasonal-latitudinal variation of density which is calculated for altitudes between 90 and 170 km in subroutine SLV). In the J70MM model the inclusion of seasonal- latitudinal variations at altitudes above 500 km led to a discontinuity in the helium number densities (and also, therefore, in the total mass density) across the 500 km level. This undesirable feature of the J70MM model has been eliminated in the MET model by the use of subroutine FAIR5 [7]. FAIR5 is invoked between 440 and 500 km altitude, while SLVH is now invoked at all altitudes above 440 km (instead of 500 km). Using a spline routine, FAIR5 adds progressively more and more of the seasonal-latitudial variation to the helium number densities as one progresses from 440 to 500 km altitude, so that no effect of SLVH is included at 440 km altitude, but the full affect of SLVH is included at 500 km altitude. While FAIR5 now corrects the density discontinuity problem at 500 km altitude, it does not in any way significantly effect the total mass density between the altitudes of 440 and 500 km, because helium is still a minor species in this altitude range.

SUBROUTINE J70SUP.

This subroutine calculates supplementary atmospheric parameters which are not directly calculated in J70. The input parameters for J70SUP are altitude and the entire OUTDATA array from J70. The output parameters calculated by J70SUP are g (gravitational acceleration), H (pressure scale-height), Cp (specific heat at constant pressure), C_v (specific heat at constant volume) and γ (ratio C_p/C_v).

For consistency g is calculated as in [1], given by

$$g = 9.80665\ (1 + Z/R_e)^{-2}\ \mathrm{ms}^{-2} \qquad (5)$$

where Z is altitude (in km) and the Earth's radius $R_e = 6.356766 \times 10^3$ km. The equation for the pressure scale-height, H, can be found in most standard texts (eg.[6]) and in a hydrostatic and

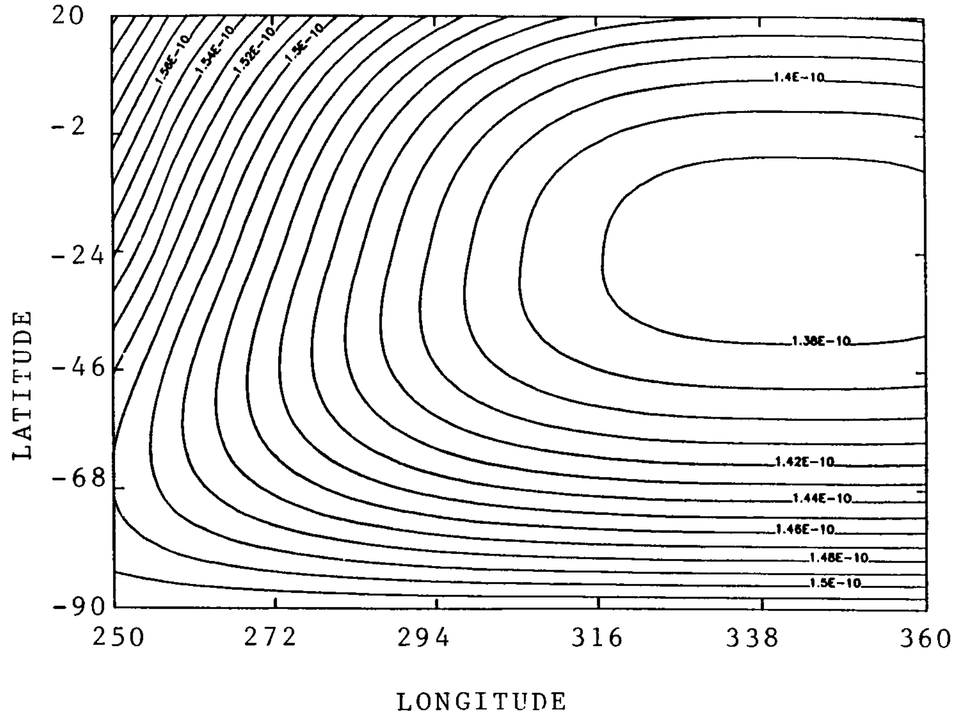

Figure 3(a) Contour plot of MET model mass density

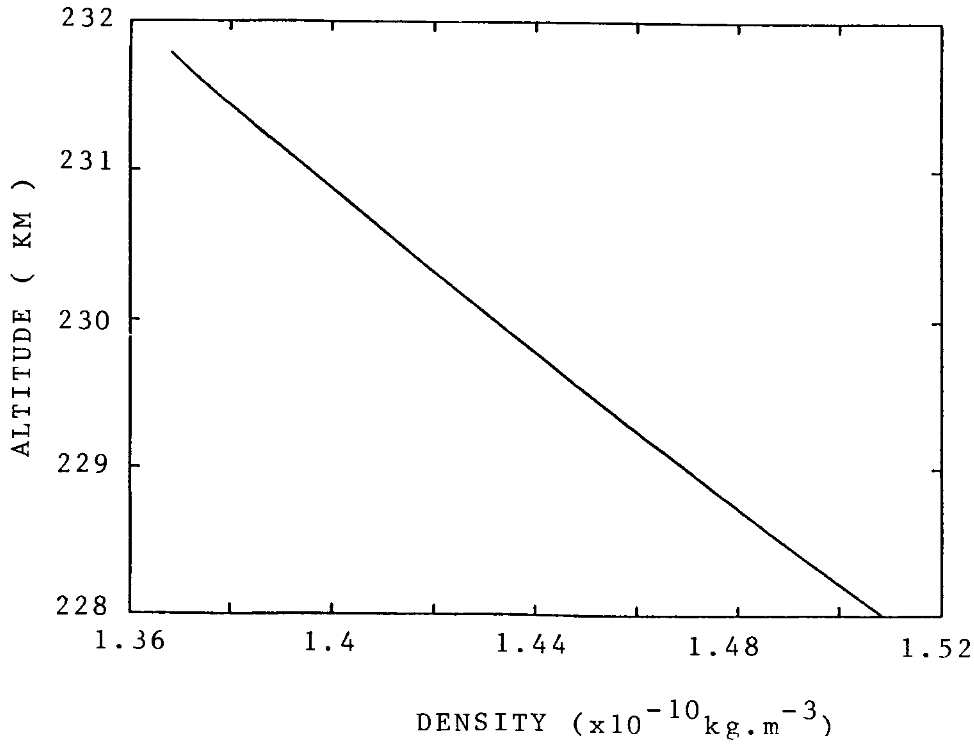

Figure 3(b) Altitude plot of MET model mass density isothermal atmosphere is given by

$$H = p/\rho g \tag{6}$$

where p is pressure and ρ is mass density. The value of γ is then calculated from the weighted mean of all diatomic and monatomic gases by assuming that all diatomic gases have $\gamma = 1.4$ and all monatomic gases have $\gamma = 1.67$. By using equation (6), the equation of state and the fact that $R = c_p - c_v$ (R is the gas constant) one obtains

$$C_v = gH/(\gamma-1)T \tag{7}$$

where T is temperature.

TEST CASE

The following is a test case for the MET model. For the following input:

Time: 1987, October 27, 1400 hrs
Indices: $F_{10.7} = 230$, $F_{10.7} = 230$, AP = 400

Altitude = 200 km, Latitude = Longitude = 0^o

The following output should obtain:

Exospheric temperature	= 2105.903 K
Local temperature	= 1495.590 K
N_2 number density	= 6.378800×10^{15} m^{-3}
O_2 number density	= 7.158251×10^{14} m^{-3}
O number density	= 4.897181×10^{15} m^{-3}
A number density	= 1.146983×10^{13} m^{-3}
He number density	= 9.737494×10^{12} m^{-3}
H number density	= 1.00000×10^{6} m^{-3}
Average molecular weight	= 23.34522
Total mass density	= 4.656589×10^{-10} kgm^{-3}
Log_{10} (mass density)	= -12.33193
Total pressure	= 2.480329×10^{-4} Pa
Local gravitational acceleration	= 9.217513
Ratio of specific heats	= 1.510544
Pressure scale-height	= 57786.66
Specific heat at constant pressure	= 1053.729 $m^2s^{-2}K^{-1}$
Specific heat at constant volume	= 697.5829 $m^2s^{-2}K^{-1}$

For the same input conditions the original J70MM model produced an exospheric temperature of 2105.775K, a local temperature of 1495.539K and a mass density of 4.656528×10^{-10} kgm^{-3} (the entire output from J70MM is not given here). Therefore, for this particular case the differences between results from the MET model and the J70MM model are typically less than 0.01%. One must remember, however, that occasionally results from the two models will differ by more than 1% (eg. Figure 2).

CONCLUSIONS AND RECOMMENDATIONS

This report has described how the MSFC/JO model was modified in order to produce the NASA/MET model. Although this report has been primarily concerned with the several modifications which directly affect the numerical output of the MET model, most of the modifications to the MSFC/J70 model entailed extensive rewriting of the FORTRAN code with the result that the MET model, written is FORTRAN 77, is easy to follow and easy to tailor to suit ones individual needs. The MET model computer program, listed in the Appendix, should be compared with the J70MM model computer program, which is listed in [3].

Most of the numerical differences between the output of the two models is generally small, and typically fractions of a percent. Occasionally, however, differences can be about one percent (in total mass density), and these are associated with the intermittent integration error in the MSFC/J70 model. The MET model outputs several parameters which the MSFC/J70 (or J70MM) model did not. These extra parameters may be useful to some users.

The MET model is currently available to users who have access to the NASA/GSFC ENVIRONET network. In the present report, it is recommended that the MET model replace the MSFC/J70 and J70MM models for all studies relating to all design, building and operations of NASA space vehicles.

References

1. Jacchia, L. G.: New Static Models of the Thermosphere and Exosphere with Empirical Temperature Profiles. Smithsonian Astrophysical Observatory Special Report 313, May 6, 1970.

2. Jacchia, L. G.: Revised Static Models of the Thermosphere and Exosphere with Empirical Temperature Profiles. Smithsonian Astrophysical Observatory Special Report 332, May 5, 1971.

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

4. Hickey, M. P.: Driving Programs for the NASA Marshall Engineering Thermosphere Model. NASA Contract Report, in preparation, 1988.

5. Davis, T. M.: Branch Note, NASA/MSFC, ED44, 1985.

6. Batchelor, G. K.: An Introduction to Fluid Dynamics. Cambridge University Press, 1967.

7. Jeffries, W.: Branch Memo, NASA/MSFC, ED44, 1986.

APPENDIX

```
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)

      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

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

```
Type *,' Specific heat cons. v  = ', AUXDATA (5),'  m2.sec-2.K-1'
Type *,' '

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

      DATA IDAY / 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 /

      PARAMETER PI = 3.14159265

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 )

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*      --  02  ..        ..            =    ..    (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. GT. 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 )                  **
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)

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

        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 )
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
      RAS = ASIN ( TAN ( SDA ) / TAN ( B4 ) )
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 , TPI = 6.28318531
      PARAMETER XM = 2.5 , XNN = 3.0
C
C**  Ci are solar activity variation variables
C
      PARAMETER C1 = 383.0 , C2 = 3.32 , C3 = 1.80
C
C**  Di are geomagnetic variation variables
C
      PARAMETER D1 = 28.0 , D2 = 0.03 , D3 = 1.0 , D4 = 100.0 , D5 = -0.08
C
C**  Ei are semiannual variation variables
C
      PARAMETER E1 = 2.41 , E2 = 0.349 , E3 = 0.206 , E4 = 6.2831853
      PARAMETER E5 = 3.9531708 , E6 = 12.5663706 , E7 = 4.3214352
      PARAMETER E8 = 0.1145 , E9 = 0.5 , E10 = 6.2831853
      PARAMETER E11 = 5.9742620 , E12 = 2.16

      PARAMETER BETA = -0.6457718 , GAMMA = 0.7504916 , 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 )
      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 'INTEGRATE' 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.

      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. )

      FA = MOL_WT ( A ) * GRAVITY ( A ) / TEMP ( A , TX , T1 , T3 , T4 , A2 )
      GD = GRAVITY ( D )
      TD = TEMP ( D , TX , T1 , T3 , T4 , A2 )
      EM = MOL_WT ( D )
      FD = EM * GD / TD

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

      CALL INTEGRATE ( A , D , FA , FD , R , TX , T1 , T3 , T4 , A2 )

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

      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


              FA1 = GD / TD
              FD1 = GRAVITY ( Z ) / TZ

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

      CALL INTEGRATE ( D , Z , FA1 , FD1 , 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))

              FA1 = GRAVITY ( A1 ) / S

      CALL INTEGRATE ( A1 , Z , FA1 , FD1 , 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 FUNCTION MOL_WT*4 ( 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 INTEGRATE ( A , D , FA , FD , RR , TX , T1 , T3 , T4 , A2 )

C*****************************************************************************
C** Subroutine 'INTEGRATE' performs the Simpson's Rule quadrature ( SRQ4 )  **
C** implemented by G. F. Kuncir.                                            **
C**                                                                         **
C**  A    = lower limit of integration                                      **
C**  D    = upper limit of integration                                      **
C**  EPS  = relative error convergence criterion                            **
C**  M    = maximum number of integrations                                  **
C**  RR   = result of integration                                           **
C**  N    = number of integrations required to find R                       **
C**                                                                         **
C**   Written by Mike Hickey, USRA                                          **
C*****************************************************************************

      DIMENSION  B (7)
      REAL*4  MOL_WT, R ( 0:10 )

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

      M = 10
      EPS = .0001
      SONE = ( D - A ) * ( FA + FD ) * 0.5
      R (0) = 0.0

              DO  2  N = 1 , M

      NINT = 2 ** N
      STOW = 0.
      DEL  = ( D - A ) / FLOAT ( NINT )

              DO   1   I = 1 , NINT , 2

      X    = A + DEL * FLOAT ( I )

      FX = MOL_WT (X) * GRAVITY ( X ) / TEMP ( X , TX , T1 , T3 , T4 , A2 )

              STOW = STOW + FX

1             CONTINUE


      R (N) = ( SONE + 4. * DEL * STOW ) / 3.

C  For altitudes greater than 180 km N usually reaches 5 or 6 before
C  convergence is achieved.  Quite occasionally, R can appear to reach
C  convergence for N=3 at altitudes above 200 km.  This anomalous and erroneous
C  result can be avoided by forcing N to go to values greater than 3 before
C  checking for convergence at altitudes greater than 180 km :-

      IF ( ( D .GE. 180. .AND. N .GE. 4 ).OR. D .LT. 180. )  THEN
              IF ( EPS * ABS ( R(N) ). GE. ABS ( R(N) - R(N-1) ) )  THEN
                      RR = R (N)
                      RETURN
              END IF
      END IF

      SONE = 0.25 * ( SONE + 3. * R (N) )

D             IF ( N .EQ. M ) TYPE *,'WARNING : Convergence not achieved'

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.
      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 variation 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 )
      DATA CZ / 1.0, 0.9045085, 0.6545085, 0.3454915, 0.0954915, 0.0 /
      PARAMETER IBFH = 440

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
```


☆U.S. GOVERNMENT PRINTING OFFICE: 1988 - 530-050/80083