NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE
A RADIATIVE HEATING AND COOLING ALGORITHM FOR A NUMERICAL MODEL OF THE LARGE SCALE STRATOSPHERIC CIRCULATION

William M. Wahr and Conway B. Leovy

JULY 1981

SUPPORTED BY:
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Grant NAG2-68

Department of Atmospheric Sciences
University of Washington
Seattle, Washington 98195
A RADIATIVE HEATING AND COOLING ALGORITHM FOR A NUMERICAL MODEL OF THE LARGE SCALE STRATOSPHERIC CIRCULATION

WILLIAM M. WEHRBEIN AND CONWAY B. LEOVY

DEPARTMENT OF ATMOSPHERIC SCIENCES UNIVERSITY OF WASHINGTON SEATTLE, WASHINGTON 98195

JULY 1981
ACKNOWLEDGMENT

This research was supported by the Upper Atmosphere Research Program of the National Aeronautics and Space Administration under NASA Grant NAG2-66.
## CONTENTS

1. INTRODUCTION ............................................. 1

2. ATMOSPHERIC CONSTITUENTS ................................... 2
   2.1 Carbon dioxide ........................................... 2
   2.2 Ozone .................................................. 2
   2.3 Molecular oxygen ....................................... 4

3. DERIVATION OF CURTIS X TRIX ELEMENTS ..................... 5
   3.1 Vertical grid ............................................. 8
   3.2 The overlap of spectral lines ............................ 10
   3.3 Line shapes ............................................. 10
   3.4 Variation of line strength with temperature ............ 11
   3.5 Non-LTE source function ................................ 13
   3.6 Temperature dependence of transmission function .... 14

4. INFRARED COOLING BY OZONE .................................. 18

5. PROGRAMS AND SUBROUTINES ................................... 21
   5.1 Routines used to obtain line strength distributions. 21
   5.2 Routines used to obtain Curtis matrices ................. 22
   5.3 Routines used for ozone and oxygen ..................... 24
   5.4 Routines used in the dynamical model ................. 25
   5.5 Changes in main program WAVE2 .......................... 25

REFERENCES

APPENDIX: FORTRAN Code
1. INTRODUCTION

A radiative heating and cooling algorithm has been developed for use in the dynamical model of the middle atmosphere written by J. R. Holton (Holton and Wehrbein, 1979; Holton and Wehrbein, 1980). A Curtis matrix is used to compute cooling by the 15µ and 10µ bands of carbon dioxide. Escape of radiation to space and exchange with the lower boundary are used for the 9.6µ band of ozone. Voigt line shape, vibrational relaxation, line overlap, and the temperature dependence of line strength distributions and transmission functions are incorporated into the Curtis matrices. Properties of this algorithm have been outlined in Wehrbein and Leovy (1981). This report is a more detailed description of its development.

Section 2 discusses the distributions of the atmospheric constituents included in this algorithm. Section 3 describes the method used to compute the Curtis matrices. Cooling or heating by the 9.6µ band of ozone is discussed in section 4. A description of the FORTRAN programs and subroutines that have been developed in this study is given in section 5. Listings of these routines are found in the appendix.
2. ATMOSPHERIC CONSTITUENTS

With the exception of molecular oxygen, which absorbs solar radiation in the upper mesosphere, the atmospheric constituents that determine the thermal structure of the middle atmosphere are exceedingly minor constituents. A discussion of the contributions of each gas is found in London (1980). In the present work we shall consider only carbon dioxide (15μ and 10μ bands), and ozone (9.6μ band and solar absorption), and molecular oxygen (solar absorption). A more sophisticated algorithm might include the overlap of carbon dioxide and ozone bands with each other, and with the bands of tropospheric water vapor when computing radiative exchange with the ground (e.g., Ramanathan, 1976).

2.1 Carbon dioxide

Carbon dioxide is, for our purposes, mixed throughout the atmosphere with volume mixing ratio of 0.33 x 10^{-3}. Therefore the amount of CO₂ between any two pressure levels is always the same, and the transmission between two pressure levels depends only on the temperature of the intervening material.

2.2 Ozone

Both the local density and the column abundances of ozone and molecular oxygen are required to compute the radiative heating due to absorption of solar radiation. The specific ozone density \(Q_3 = \frac{n_3}{n_{STP}}\) where \(n_{STP} = 2.6869 \times 10^{19} \text{ cm}^{-3}\) and \(n_3\) is the number density of ozone. The mass mixing ratio (in grams/gram) is given by \(W_3 = \frac{\rho_3}{\rho_a}\), where \(\rho_3\) and \(\rho_a\) are the mass densities of ozone and air, respectively. The specific density is related to the mass mixing ratio by
\[ Q_3 = \frac{p}{R_a T} \frac{W_3}{m_3 n_{\text{STP}}} \]  

(2.1)

where \( p \) and \( T \) are local pressure and temperature, \( R_a \) is the gas constant for air, and \( m_3 \) is the mass of a single ozone molecule.

The specific column abundance (in cm at STP) is given as

\[ \eta_3 = \int_{z}^{\infty} Q_3(z')dz' = \frac{1}{m_3 n_{\text{STP}}} \int_{0}^{p} W_3(p')dp' \]  

(2.2)

Above a certain altitude \( z_1 \) (pressure level \( p_1 \)) we assume that ozone is distributed with a constant scale height \( H_3 \) so that

\[ n_3(z_1) = \frac{1}{n_{\text{STP}}} \int_{z}^{\infty} n(z_1) e^{-(z-z_1)/H_3} dz = Q_3(z_1)H_3 \]

Below \( p_1 \) the mean value of \( W_3 \) in each layer is used to obtain

\[ \eta_3(p_j) = \eta_3(p_1) + \frac{1}{m_3 n_{\text{STP}}} \frac{1}{2} \left( \sum_{j=1}^{J-1} \left[ W_3(p_j) + W_3(p_{j+1}) \right] (p_{j+1} - p_j) \right) \]

For arbitrary \( p \) \( \eta_3 \) is interpolated linearly between levels:

\[ \eta_3(p) = \eta_3(p_j) \gamma + (1 - \gamma) \eta_3(p_{j+1}) \] where \( \gamma = \frac{p - p_j}{p_{j+1} - p_j} \)  

(2.3)

for \( p_j \leq p \leq p_{j+1} \).
2.3 Molecular oxygen

Assume that the mixing ratio of oxygen in air is 0.2095 up to 155 km, (3.645 x 10^{-9} atmospheres), above which oxygen follows the profile

\[ n_2(z) = 1 \times 10^{17} e^{-z/30 \text{km}}. \]

Below 155 km \( n_2(z) = 0.2095 \frac{p}{k_B T} \) where \( k_B \) is Boltzmann’s constant.

The column abundance below 155 km is therefore given by

\[ \eta_2 = \int_0^z n_2(z') dz' = 1.711 \times 10^{15} \frac{0.2095}{m_a} (p - 3.645 \times 10^{-9} p_0) \]  \( (2.4) \)

where \( m_a \) is the mean mass of a molecule of air, \( p_o \) is standard surface pressure, and \( \eta_2 \) is given in molecules cm^{-2}. 
3. DERIVATION OF CURTIS MATRIX ELEMENTS

The net upward flux of thermal radiation at wavenumber $\nu$ is given by

$$F_{\nu}(\tau) = 2 \pi \left[ J_{\nu}^+(p) \exp \left\{ - \int_{k_{\nu} c dp''}^{p'} \frac{cdp'''}{g\mu} \right\} k_{\nu} \frac{c}{g} dp' \right]_{p}^{p_s}$$

$$- 2 \pi \int_{0}^{1} d\mu \int_{0}^{\infty} J_{\nu}^{-}(p') \exp \left\{ - \int_{k_{\nu} c dp''}^{p'} \frac{cdp'''}{g\mu} \right\} k_{\nu} \frac{c}{g} dp'$$

$$(Goody, 1964, p. 49)$$

where $\mu$ is the absolute value of the cosine of the zenith angle, and the optical depth is defined as

$$\tau = \tau(p) = \frac{1}{g} \int_{0}^{p} k_{\nu}(p'')c(p'')dp''$$

$$= \int_{0}^{1} d\mu \int_{0}^{\infty} J_{\nu}(p) \exp \left\{ - \int_{k_{\nu} c dp''}^{p'} \frac{cdp'''}{g\mu} \right\} k_{\nu} \frac{c}{g} dp'$$

where $k_{\nu}$ is the absorption coefficient and $c$ the concentration of the absorbing gas. The source function $J_{\nu}$ is assumed to be isotropic in the upper (+) and lower (-) hemispheres. Changing the variable of integration to pressure yields

$$F_{\nu}(p) = 2 \pi \int_{0}^{1} d\mu \int_{0}^{p_s} J_{\nu}^+(p) \exp \left\{ - \int_{k_{\nu} c dp''}^{p'} \frac{cdp'''}{g\mu} \right\} k_{\nu} \frac{c}{g} dp'$$

$$+ 2 \pi \int_{0}^{1} d\mu \int_{p_s}^{p} J_{\nu}^+(p) \exp \left\{ - \int_{k_{\nu} c dp''}^{p'} \frac{cdp'''}{g\mu} \right\} k_{\nu} \frac{c}{g} dp'$$

$$- 2 \pi \int_{0}^{1} d\mu \int_{p}^{p_s} J_{\nu}^{-}(p) \exp \left\{ - \int_{k_{\nu} c dp''}^{p'} \frac{cdp'''}{g\mu} \right\} k_{\nu} \frac{c}{g} dp', (3.3)$$
where \( p_a \) is the surface pressure. The second term, arising from the surface, can be integrated over \( p' \) directly.

Consider the flux of radiation integrated over a frequency interval \( \Delta \nu \) sufficiently small that \( J^+_{\nu} \) can be approximated by mean values over frequency \( J^+_{\nu} \). Then changing the order of integration gives

\[
\int \frac{d\nu}{\Delta \nu} F^+_{\nu}(p') = 2\pi \int_{p}^{p_a} dp' J^+_{\nu}(p') \int_{0}^{1} d\mu \int_{\Delta \nu} dv \exp \left\{ - \int_{p}^{p'} k_{\nu} \frac{cdp''}{g\mu} \right\} \frac{k_{\nu}}{g} \]

\[
+ 2\pi J^+_{\nu}(p_a) \int_{0}^{1} d\mu \int_{\Delta \nu} dv \exp \left\{ - \int_{p}^{p_a} k_{\nu} \frac{cdp''}{g\mu} \right\} \]

\[
- 2\pi \int_{p}^{p_a} dp' J^-_{\nu}(p') \int_{0}^{1} d\mu \int_{\Delta \nu} dv \exp \left\{ - \int_{p}^{p'} k_{\nu} \frac{cdp''}{g\mu} \right\} \frac{k_{\nu}}{g} \quad . \tag{3.4}
\]

Note that each of the integrals in the exponents are non-negative.

Define the flux equivalent width

\[
e(p,p') = \int_{0}^{1} d\mu \int_{\Delta \nu} dv [1 - T_{\nu}(p,p',\mu)]
\]

where

\[
T_{\nu}(p,p',\mu) = \exp \left\{ - \int_{p}^{p'} k_{\nu} c/g\mu \frac{dp''}{g} \right\} \quad . \tag{3.5}
\]

Function \( e(p,p') \) is simply one minus the transmission function between \( p \) and \( p' \) weighted by \( \mu \) integrated over one hemisphere. Clearly \( e(p,p) = 0 \), and
\[
\frac{\partial \epsilon}{\partial p} = \int_0^1 d\mu \int_{\Delta \nu} d\nu \exp \left\{ - \int_p^{p'} \frac{k_c dp'}{\Delta \nu} \right\} \frac{k_c}{\Delta \nu} \times \begin{cases} 1, & p' > p \\ \nu(p'), & p' < p \end{cases} .
\] (3.6)

Changing variables from \( p \) to \( \epsilon \), the expression for the integrated flux becomes

\[
\int_{\Delta \nu} d\nu F^{+}(\nu, p) = 2\pi \int_0^\epsilon \frac{e(\nu, p, p')}{\Delta \nu} \left[ J^+(p') de(p, p') + 2\pi J^+(p_s) - e(p, p_s) \right]

+ 2\pi \int_0^\epsilon J^-(p') de(p, p') .
\] (3.7)

Define the upward flux

\[
F^+(p) = 2\pi \int_0^\epsilon \frac{e(\nu, p, p')}{\Delta \nu} \left[ J^+(p') de(p, p') + 2\pi J^+(p_s) - e(p, p_s) \right]
\]

and the downward flux

\[
F^-(p) = 2\pi \int_0^\epsilon \frac{e(\nu, p, p')}{\Delta \nu} J^-(p') de(p, p') .
\] (3.8)

Assume the source function is isotropic over all directions \( J^+ = J^- \), and define a nondimensional "temperature" \( \theta = J/\bar{B} \), where \( \bar{B} \) is the Planck function in \( \Delta \nu \) evaluated at some reference temperature. We have

\[
\int_{\Delta \nu} d\nu F^+(\nu, p) - F^-(p) = 2\pi \bar{B} \int_0^\epsilon \theta(p') de(p, p')

+ 2\pi \bar{B} \left[ \frac{\Delta \nu}{2} - e(p, p_s) \right] - 2\pi \bar{B} \int_0^\epsilon \theta(p') de(p, p') .
\] (3.9)
3.1 Vertical grid

Divide the atmosphere into \( N \) layers (see Fig. 1). The pressure at \( \phi_1 \) is zero, and \( \phi_{N+1} \) is at the ground. Pressure and temperature evaluated at levels within layers are denoted with carets. To simplify notation, let

\[
e(p_i, p_j) = e_{ij} \quad e(p_i, \hat{p}_j) = \hat{e}_{ij} \quad e(p_i, p_s) = e_{i,N+1}.
\]

For the simplest case \( \beta \) is treated as a constant throughout each layer. Then

\[
F_i^+ = 2\pi B \sum_{j=1}^{N} (e_{i,j+1} - e_{ij}) \hat{\theta}_j + 2\pi B \left( \frac{\Delta \nu}{2} - e_{i,N+1} \right) \hat{\theta}_{N+1}
\]

\[
= 2\pi B \sum_{j=1}^{N+1} (e_{i,j+1} - e_{ij}) \hat{\theta}_j \quad \text{where} \quad e_{i,N+2} \equiv \Delta \nu / 2, \quad (3.10)
\]

and

\[
F_i^- = -2\pi B \sum_{j=1}^{i-1} (e_{i,j+1} - e_{ij}) \hat{\theta}_j.
\]

The convergence of radiant flux in layer \( i \) is given by

\[
\Delta F_i = F_{i+1}^+ - F_{i+1}^- - F_i^+ - F_i^-
\]

\[
= 2\pi B \sum_{j=1}^{N+1} (e_{i+1,j+1} - e_{i+1,j} - e_{i,j+1} + e_{ij}) \hat{\theta}_j. \quad (3.11)
\]

The change in temperature in layer \( i \) resulting from the convergence of radiant flux is
\[ q_i = \sum_{j=1}^{N+1} R_{ij} \dot{\theta}_j \quad \text{where} \quad R_{ij} = \frac{2\pi B g}{\Delta p_i c_p} \left( \epsilon_{i+1,j+1} - \epsilon_{i+1,j} - \epsilon_{i,j+1} + \epsilon_{ij} \right) \]  \tag{3.12}

where \( \Delta p_i \) is the pressure difference across layer \( i \).

A more complex integration scheme results if we assume that temperature varies linearly in log pressure from layer midpoint to layer midpoint:

\[
\begin{align*}
\theta(\phi) &= \gamma_1 \hat{\theta}_1 + (1 - \gamma_1) \hat{\theta}_{i-1}, \\
\hat{\phi}_{i-1} &\leq \phi \leq \hat{\phi}_i
\end{align*}
\]

where \( \gamma_1 = (\ln \phi - \ln \hat{\phi}_{i-1})/(\ln \hat{\phi}_1 - \ln \hat{\phi}_{i-1}) \). Above \( \hat{\phi}_1 \) the atmosphere is assumed to be isothermal at \( \theta_1 \), and below \( \hat{\phi}_N \) the temperature is extrapolated from level \( \hat{\phi}_{N-1} \) to a temperature discontinuity at the surface. The expression for the Curtis matrix elements becomes

\[ R_{ij} = \frac{2\pi B g}{\Delta p_i c_p} \left( \epsilon'_{i+1,j+1} - \epsilon'_{i+1,j} - \epsilon'_{i,j+1} + \epsilon'_{ij} \right), \]

where \( \epsilon'_{ij} = \epsilon_{ij} + c_{ij} \) and correction term \( c_{ij} \) is given by

\[
\begin{align*}
c_{ij} &= \begin{cases} 
0, & \text{if } j = 1 \\
\hat{\epsilon}_{ij} - \epsilon_{ij} - \begin{cases}
\epsilon_{i,j-1}, & 2 \leq j \leq N-1 \\
\epsilon_{i,N-1} - \epsilon_{i,N}, & j = N
\end{cases}, & j = N+1 \text{ or } N+2
\end{cases}
\end{align*}
\tag{3.13}
\]

The integrals in (3.13) have been performed numerically using the trapezoidal rule.
3.2 The overlap of spectral lines

Below about 30 km pressure-broadened line widths may be sufficiently large to cause significant overlapping of individual rotational lines. The entire band (550-800 cm\(^{-1}\)) is divided into ten equal spectral subintervals, and spectral lines are assumed to be randomly distributed in position within each subinterval. Eqn. (3.5) can be written

\[ e(p_i, p_j) = \Delta \nu \left[ \frac{1}{2} - \int _0 ^{1} d \mu \frac{1}{\Delta \nu} \sum _{k=1} ^{10} \delta \nu _k \overline{T} _k \right], \quad (3.14) \]

where \( \overline{T} _k = \frac{1}{\delta \nu _k} \int d \nu \overline{T} _\nu \) is the mean transmission function over subinterval \( k \).

The assumption of random distribution implies (Goody, 1964, p. 154)

\[ \overline{T} _k = \exp \left( - \frac{1}{\delta \nu _k} \sum _{k} \overline{w} _k \right), \quad (3.15) \]

where \( \overline{w} _k \) is the equivalent line width of the \( k \)th line in subinterval \( k \).

3.3 Line shapes

Through much of the region of interest the shape of the individual spectral lines is given by the Voigt profile, a convolution of the simpler Doppler and Lorentz shapes. The equivalent width of a Voigt line can be adequately approximated with a formula that interpolates between the equivalent line widths given by the Doppler and Lorentz profiles (Rodgers and Williams, 1974)

\[ W(p_i, p_j, \nu) = (\overline{w} _D ^2 + \overline{w} _L ^2 - \overline{w} _D ^2 \overline{w} _L ^2 / S ^2 \nu ^2)^{1/2}, \quad (3.16) \]
where

\[ \omega_L = \omega_L([u/\mu,S(\tilde{T}_{ij}),\alpha_L(\tilde{P}_{ij},\tilde{T}_{ij})]) \]

and

\[ \omega_D = \omega_D([u/\mu,S(\tilde{T}_{ij})]) \]

\( \omega_L \) is the Lorentz equivalent line width given by an accurate approximation due to Goldman (1968). \( \omega_D \) is the Doppler equivalent line width given by a series approximation or a large \( S_u \) approximation (Goody, 1964, p. 136).

\( S(\tilde{T}_{ij}) \) is the line strength evaluated at the Curtis-Godson temperature \( \tilde{T}_{ij} \) between pressure levels \( p_i \) and \( p_j \) and \( u \) is the amount of absorber (CO\(_2\)) between levels \( p_i \) and \( p_j \). The Lorentz line width \( \alpha_L(\tilde{P}_{ij},\tilde{T}_{ij}) \) is given by

\[ \alpha_L(\tilde{P}_{ij},\tilde{T}_{ij}) = 0.98 \text{ cm}^{-1}\left(\frac{\tilde{P}_{ij}}{p_o}\right)\left(\frac{300 \text{ K}}{\tilde{T}_{ij}}\right)^{1/2} \]

where the Curtis-Godson parameters are

\[ \tilde{T}_{ij} = \frac{1}{p_i - p_j} \int_{p_j}^{p_i} T(p) \, dp \text{ and } \tilde{P}_{ij} = \frac{1}{2} (p_i + p_j) \]  

For a given temperature profile the Curtis-Godson temperature was computed for every pair of levels and stored for future reference as a two-dimensional array.

### 3.4 Variation of line strength with temperature

There are nearly 17,000 CO\(_2\) lines tabulated between 550 and 800 cm\(^{-1}\) on the AFGL Atmospheric Absorption Line Parameter Compilation ("McClatchey tape"). Rather than sum over all these lines in (3.15), the lines in each subinterval were divided into groups according to line strengths, and a mean line strength was used to represent the strength of all the lines in
that group. Working with one subinterval at a time, the strength of each line was computed at seven temperatures between 150 K and 296 K with the formula

$$S(T) = \frac{S(T_S)Q_v(T_S)Q_r(T_S)}{Q_v(T)Q_r(T)} \exp \left( \frac{1.439E''(T-T_S)}{TTS} \right),$$  \hspace{1cm} (3.17)

where $Q_v$ and $Q_r$ are the vibrational and rotational partition functions interpolated from the table in McClatchey et al. (1973). $T_S$ is a reference temperature, and $E''$ is the energy of the lower state of the vibrational-rotational transition expressed in $\text{cm}^{-1}$. $S$ is expressed in the units $\text{cm}^{-1} (\text{cm STP})^{-1}$. At each temperature the lines were sorted into 50 groups representing equal increments of $\log_{10} S$. The resulting histograms of line strengths for one subinterval are shown in Fig. 2. The histograms were represented approximately by dividing the lines into two categories: a set of temperature-independent strong lines, and a larger set of weaker lines which follow the distribution function

$$n_i(x) = m \ [x - x_{0i}]^2 \ \text{for} \ x < x_{0i},$$  \hspace{1cm} (3.18)

where $x = \log_{10} S$. Here $x_{0i} = \log_{10} S_{0i}$ where $S_{0i}$ is the strength corresponding to the empty bin of lines just stronger than the strongest temperature dependent lines in subinterval $i$. Factor $m$ was determined for each temperature in each subinterval, but was found to be adequately represented by a universal function

$$m(T) = aT + b, \ a = 0.001615, \ b = -0.2066,$$  \hspace{1cm} (3.19)
where \( a \) and \( b \) are evaluated by regression. The mean strength in each group is given by
\[
\overline{S}_j = \exp(2.303 \overline{x}_j) \text{ where } \overline{x}_j = \int dxn(x)x/\int dxn(x),
\]
(3.20a)
and the number of lines having this strength is
\[
F_j = \int dxn(x).
\]
(3.20b)

The number and mean strength of the strong lines in each subinterval was evaluated directly. These strong lines, together with weak lines evaluated from (3.20) were used to compute mean transmission functions, flux equivalent widths, and finally, the Curtis matrix elements for a standard temperature profile.

3.5 Non-LTE source function (this section follows Houghton, 1977, p. 58ff)

Above about 65 km the frequency of collisions is insufficient to maintain a population of excited vibrational states consistent with the local kinetic temperature. The source function for a particular transition is no longer identical to the Planck function, but is given by
\[
J_v = \frac{\overline{L}_v + \epsilon B_v}{1 + \epsilon},
\]
(3.21)
where \( \epsilon = C/A \) is the ratio of the rate of collisional deactivation to the rate of spontaneous radiative deactivation of the excited state. Assuming \( J_v \) and \( B_v \) are constant over the frequency band, and using the equation of radiative transfer to eliminate the mean radianc \( \overline{L}_v \) gives
\[
J = B + q \epsilon \sigma T^4 S_b,
\]
(3.22)
where \( c \) is the concentration of the absorber, and \( S_b \) and \( q \) are the local band strength and radiative heating of the particular transition. Eqn. (3.12) must be modified to

\[
q_i = \sum_{j=1}^{N+1} R_{ij} \left( \frac{J_j}{\bar{E}} \right)
\]  

(3.23)

Using matrix notation, (3.22) and (3.23) can be combined into

\[
q = R^\text{LTE} \quad \text{where} \quad R^\text{LTE} = (\hat{c} - \bar{E})^{-1} R
\]

(3.24)

and

\[
E_{ij} = \left[ \frac{c_p}{4\pi^2 E_b} S_{b,i} \right] \delta_{ij}
\]

(3.25)

\( S_{b,i} \) and \( \varepsilon_i \) are the band strength and LTE factor evaluated in the \( i \)th layer.

In the present calculation all lines are considered part of the fundamental band with Einstein coefficient \( A = 0.74 \text{s}^{-1} \) and collision frequency \( C = 0.67 \times 10^5 \text{ s}^{-1} \) at standard pressure and an assumed mesopause temperature of 180K. The band strength \( S_{b,i} \) is evaluated as

\[
S_{b,i} = \sum_{k=1}^{10} \left( \sum_j F_{kj} \overline{S}_{kj} + F_k^S S^S \right)
\]

(3.25)

where \( F_{kj} \) and \( \overline{S}_{kj} \) are given by (3.20) and \( F^S \) and \( S^S \) refer to the strong lines in subinterval \( k \).

3.6 Temperature dependence of transmission function

Since the flux equivalent widths depend on the temperature of the intervening material, the Curtis matrix elements are functions of the
atmospheric temperature profile. The flux equivalent width is related to
the transmission function, so we have adapted the interpolation method of
Fels and Schwarzkopf (1981).

Let \( k_v(p'') = \sum_k f_k(v, T_0(p''), p'') \) where \( f_k \) is the absorption coeffi-
cient of the \( k \)th line in \( \delta v \). Let \( T(p'') = T_0(p'') + \delta T(p'') \) where \( T_0(p'') \)
is the standard temperature profile. To first order in \( \delta T \)

\[
T_v(p, p', \mu) = \exp \left[ - \int_p^{p'} \sum_k f_k(v, T_0(p''), p'') \frac{\partial f_k}{\partial T} \bigg|_{T_0} \delta T(p'') \right] \frac{c d p''}{g \mu}
\]

\[
= \exp \left[ - \int_p^{p'} \sum_k f_k(v, T_0(p''), p') \frac{c d p''}{g \mu} \right] \exp \left[ - \int_p^{p'} \sum_k \frac{\partial f_k}{\partial T} \bigg|_{T_0} \delta T(p'') \frac{c d p''}{g \mu} \right] .
\]

Assume \( \delta T(p'') \) is small and expand the second exponent

\[
T_v(p, p', \mu) = T_0^v(p, p', \mu) \left[ 1 - \int_p^{p'} \sum_k \frac{\partial f_k}{\partial T} \bigg|_{T_0} \delta T(p'') \frac{c d p''}{g \mu} \right],
\]

where

\[
T_0^v(p, p', \mu) = \exp \left[ - \int_p^{p'} \sum_k f_k(v, T_0(p''), p') \frac{c d p''}{g \mu} \right].
\]

Eqn. (3.5) becomes

\[
e(p, p') = \int_0^1 d\mu u \int_\Delta v \left[ 1 - T_0^v(p, p', \mu) \right] - \int_p^{p'} \frac{c d p''}{g} \int_0^1 d\mu \int_\Delta v T_0^v(p, p', \mu) \sum_k \frac{\partial f_k}{\partial T} \bigg|_{T_0} \delta T(p'').
\]
Now define \( e^0(p,p') = \int_0^1 d\mu \int_0^\Delta \nu [1 - T^0_v(p,p',\mu)] \) and

\[
\frac{\delta e(p,p',p'')}{\delta T} = -\frac{c}{\delta} \int_0^1 d\mu \int_0^\Delta \nu T^0_v(p,p',\mu) \sum \frac{\delta F}{\delta T} \left[ \nu, T(p'''), p''\right] \bigg|_{T_0},
\]

so

\[
e(p,p') = e^0(p,p') + \int_p^p dp'' \frac{\delta e(p,p',p'')}{\delta T} \delta T(p'').
\]

Consider the functional derivative \( \delta e/\delta T \). Suppose it can be factored (and in several approximations it can) as

\[
\frac{\delta e(p,p',p'')}{\delta T} = F(p,p')G(p''),
\]

then

\[
\delta e(p,p') = \int_p^p dp'' \frac{\delta e(p,p',p'')}{\delta T} \delta T(p'') = \int_p^p dp'' F(p,p')G(p'')\delta T(p'')
\]

\[
= F(p,p') \int_p^p dp'' G(p'')\delta T(p'')
\]

If we let \( e^+(p,p') \) represent the flux equivalent width for a temperature profile everywhere warmer than \( T_0 \) by a constant amount \( \Delta \), and similarly for \( e^-(p,p') \), then from a Taylor series we have for any constant temperature deviation \( \delta T \)

\[
e(p,p') = e^0(p,p') + \frac{e^+(p,p') - e^-(p,p')}{2\Delta} \delta T + \frac{1}{2} \left( \frac{e^+(p,p') - 2e^0(p,p') + e^-(p,p')}{\Delta^2} \right)(\delta T)^2
\]

\[
(3.26)
\]
Now define the weighted temperature deviation
\[ \Delta(p, p') = \int_{p}^{p'} dp'' G(p'') \delta T(p'') / \int_{p}^{p'} dp'' G(p'') \]  \hspace{1cm} (3.27)

We will assume that for variable temperature deviations (3.26) holds with \( \Delta(p, p') \) replacing \( \delta T \). The best choice for weighting function \( G(p) \) and for \( \Delta \) have been determined by Fels and Schwarzkopf as

\[ G(p) = p^{0.2} [1 + (p/p_0)^{0.8}] \]  \hspace{1cm} (3.28)

where \( p_0 = 30 \text{ mb} \), and \( \Delta = 25 \text{K} \).

Therefore the procedure adopted has been to compute a matrix of \( \varepsilon \), \( R \), and \( R_{nLTE} \) for each of three temperature profiles. Each \( R_{nLTE} \) is then "unraveled" using (3.12) to obtain a matrix \( \varepsilon_{nLTE} \). This can be accomplished because there are "boundary conditions" \( e_{ii} = 0 \) and \( e_{i,N+2} = \Delta \nu/2 \). Beginning with the lowest level, successive layers of \( e_{ij} \) can be computed from (3.12), first on the lower side of the diagonal \( (j > i) \), and then on the upper \( (j < i) \). These three arrays are stored as data for the dynamical model. When the radiative algorithm is called, \( \varepsilon_{nLTE}(T) \) is computed by (3.26) for the appropriate temperature profile, then \( R_{nLTE}(T) \) is computed from (3.12).

The Curtis matrix for the 10\( \mu \) band of CO\(_2\) is computed using the same procedure. The only difference is that the 10\( \mu \) band was not divided into subintervals to compute the mean transmission functions of (3.14).
4. INFRARED COOLING BY OZONE

The version of the algorithm currently incorporated in the dynamical model uses a cool-to-space plus cool-to-ground approximation utilizing the band absorption formula of Ramanathan (1976). In this section we shall develop a more general method from which the above approximation can be easily derived.

Several empirical and semiempirical formulae (e.g., Walshaw, 1957, Aida, 1975), have been developed for the integrated absorption $A$ of the 9.6$\mu$ band of ozone where

$$A = \int \frac{\Delta v A_v}{\Delta v} = \int \frac{\Delta v (1 - T_v)}{\Delta v}$$

These formulae may be used to compute flux equivalent width and Curtis matrix elements from (3.5) and (3.12). The procedure outlined in section 3 may be greatly simplified. IR cooling (or heating) by ozone is significant only below about 60 km, so non-LTE effects can be ignored. The contribution of the 9.6$\mu$ band to the total cooling is small enough that variations of the transmission function with temperature profile can be neglected. However, since the vertical distribution of ozone varies, in general, with latitude and time, a new Curtis matrix must be computed for every ozone profile.

Use of the Curtis matrix method facilitates the identification of the various components of the radiative interactions. The radiative cooling is given by

$$q_i = \sum_{j=1}^{N+1} R_{ij} \theta_j$$
where $N+1$ is the ground. If the atmosphere is isothermal, and the ground
is at the same temperature, then radiation exchanged with the atmosphere
and ground provides no net cooling and there is only cooling to space. So

$$q_{N+1}^{\text{CTS}} = \sum_{j=1}^{N+1} R_{1j} \theta_{1j} = \theta_{1} \sum_{j=1}^{N+1} R_{1j} = \theta_{1} C_{1}^{g} \quad (4.2)$$

If the ground is at a different temperature, there can be cooling to both space
and ground. Then

$$q_{N+1}^{\text{CTS}} + q_{i}^{\text{CTG}} = \sum_{j=1}^{N+1} R_{1j} \theta_{j} = \theta_{1} \sum_{j=1}^{N} R_{1j} + R_{1N+1} \theta_{N+1}$$

$$= \theta_{1} C_{1}^{g} + (\theta_{N+1} - \theta_{1}) C_{1}^{g} \quad (4.3)$$

where $C_{i}^{g} = R_{1i}, N+1$.

A further approximation can be introduced for $\theta$. The "temperature" is
given by

$$\theta = \frac{e^{u} - 1}{e^{T_{o}/T} - 1} \quad \text{where } u = h\nu/k_{o} = 4.988 \text{ for } T_{o} = 300K.$$

Hence, $e^{u} - 1 \approx e^{u}$ and for $T \leq T_{o}$ we have $e^{uT_{o}/T} - 1 \approx e^{uT_{o}/T}$,

and $\theta \approx e^{-uT_{o}/T} \cdot e^{-u}$.

So $q_{1}^{\text{CTS}} + q_{i}^{\text{CTG}} \approx u_{c}^{g} e^{-uT_{o}/T} + u_{c}^{g} (e^{-uT_{o}/T_{g}} - e^{-uT_{o}/T_{l}})$

$$= 146.6 \ C_{1}^{g} e^{-1496/T} + 146.6 \ C_{1}^{g} (e^{-1496/T_{g}} - e^{-1496/T_{l}}), \quad (4.4)$$
where $T$ is given in Kelvins. The radiative cooling of the 9.6$\mu$m band for the U.S. Standard ozone and temperature profiles computed with the Curtis matrix method employing Aida's parameterization is compared with the cool-to-space plus cool-to-ground approximation in Fig. 3.
5. PROGRAMS AND SUBROUTINES

5.1 Routines used to obtain line strength distributions

The 16899 lines attributed to carbon dioxide between 550 and 800 cm\(^{-1}\) have been sorted into eight sets according to the energy of the lower state of the transition. Each set is written on an element file of the fileset CO2SET. The description of each element set is as follows ("normal" refers to \(^{12}\text{C}^{16}\text{O}_2\)):

<table>
<thead>
<tr>
<th>File name</th>
<th>Lower level</th>
<th>Number of lines</th>
<th>Type of CO(_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU626</td>
<td>0</td>
<td>153</td>
<td>&quot;normal&quot;</td>
</tr>
<tr>
<td>HO626</td>
<td>(v_2)</td>
<td>546</td>
<td>&quot;normal&quot;</td>
</tr>
<tr>
<td>HT626</td>
<td>(v_1=2v_2)</td>
<td>159</td>
<td>&quot;normal&quot;</td>
</tr>
<tr>
<td>HD626</td>
<td>(2v_2)</td>
<td>691</td>
<td>&quot;normal&quot;</td>
</tr>
<tr>
<td>HF626</td>
<td>(3v_2)</td>
<td>564</td>
<td>&quot;normal&quot;</td>
</tr>
<tr>
<td>HC626</td>
<td>(\geq 4v_2)</td>
<td>3805</td>
<td>&quot;normal&quot;</td>
</tr>
<tr>
<td>FUISO</td>
<td>0</td>
<td>1269</td>
<td>isotopic</td>
</tr>
<tr>
<td>HTISO</td>
<td>(\geq v_2)</td>
<td>9712</td>
<td>isotopic</td>
</tr>
</tbody>
</table>

Program BINS sorts all of these lines into bins according to strength without regard to band of origin. The entire band is divided into ten equal subintervals and the strength of each line is computed at seven different temperatures. The width of each bin is 0.25 in units of log S, where S is expressed in (cm\(^{-1}\)) (cm STP\(^{-1}\)). An example of the resulting histograms is shown in Fig. 2. If the McClatchey tape is used as the input for program BINS, then DO loop 12 should be replaced with IB = 8, the two IF statements after statement number 98 should be activated, and LMAX must be set equal to or greater than 16899.
Program FIT fits each of the seventy histograms produced by BINS with a second order polynomial, according to (3.18). The only free parameter is the coefficient \( m \).

Graphs of \( m \) vs. \( T \) for each of the ten subintervals are shown in Fig. 4. It is sufficiently accurate to assume a universal function \( m(T) = aT + b \) for all subintervals. Program REGRESS provides the least square values for \( a \) and \( b \).

Spectral lines associated with the fundamental have no vibrational temperature dependence and are usually strong. They can be identified on the histograms because they rise above the generally parabolic shape of the line strength distribution profile (see Fig. 2). Program RESID subtracts the fitted line strength distribution from the actual distribution, and computes the number of strong lines and their mean strength in each bin, and then consolidates each four bins to give the number of strong lines and their mean strength in each strength decade. From this information one characteristic strong line number and strength is chosen for each subinterval, and specified in program CURTEL in arrays FS and SS.

5.2 Routines used to obtain Curtis matrices

Program CURTEL computes the Curtis matrix elements for the 15\( \mu \) band of carbon dioxide. Wave numbers are expressed in units of the Doppler width at 300K.

The computation of the non-LTE Curtis matrix begins with the evaluation of the total band strength by summing all the mean strengths weighted by the number of lines at that strength over all strength decades and subintervals. VMULFM and LINV1F are IMSL routines for matrix multiplication and inversion.
Subroutine FLUXEL calculates the equivalent band widths. First the pressure levels (PHI) and layer midpoints (PHIH) are computed. The Curtis-Godson temperature (TCG) between any two levels is computed and stored. The correction terms $c_{ij}$ [see (3.13)] are evaluated by the trapezoidal rule with the interval divided into $M_1$ panels for the diagonal terms, $M$ panels for the nondiagonal terms, and $3M/2$ panels for the terms at the top of the atmosphere.

In subroutine EFUNC there is an opportunity to use the integrated absorption parameterizations of Pollack et al. (1981) or Rodgers and Walshaw (1966) instead of the line strength distributions that have been derived. If the integrated absorption parameterizations are not chosen, the Curtis-Godson temperature and pressure are evaluated and the line strength distribution parameter [$m_{12}$ (3.18), called A2 in this subroutine] is calculated.

Subroutine W performs the summation in (3.14), evaluating the mean transmission function for the entire band. The line strength distribution parameter is used to compute the number of lines and mean strengths in nine decades of line strength for each of the ten subintervals. This synthetic line distribution is cut off at the low strength end so that the total number of lines in each subinterval is the same for the synthetic and real distributions.

After the set of non-LTE Curtis matrices have been produced for the three temperature profiles and for both 10μ and 15μ bands, program EQWD inverts (3.12) to obtain a set of equivalent band widths. The set of six arrays of $e^{nLTE}$ are used by subroutine CURT to compute the temperature-dependent Curtis matrix elements.
There are, unfortunately, two different conventions for the labeling of the layers of the model. In the dynamical model (JRH) the lowest stratospheric layer is denoted \( j = 1 \), and the highest layer is \( j = JML \) (i.e., 16). In the radiation algorithm (CBL) the highest layer is denoted \( j' = 1 \), the lowest stratospheric layer is \( j' = JML \), the troposphere is \( j' = JML + 1 \), and the ground is \( j' = JML + 2 \) (i.e., 18). Comment cards in the listing noting the shift of conventions will hopefully prevent confusion.

5.3 Routines used for ozone and oxygen

Program OZPRO produces the arrays needed to compute the radiative heating and cooling due to ozone, given the ozone mixing ratio and temperature profiles. Subroutine COLOZ computes the specific density and column abundance of ozone. Subroutine CMOZ96 computes a Curtis matrix for the 9.6\( \mu \) band, and from it cool-to-space and cool-to-ground coefficients are derived. Two different parameterizations (Aida, 1975 and Walshaw, 1957) of the integrated absorption of the 9.6\( \mu \) band are offered in function OZ96A.

Subroutine OXY computes the density and column abundance of molecular oxygen.

Function OZUV computes the solar energy absorbed per molecule of \( O_2 \) from the parameterization by Strobel (1978).

Function OZUV computes the derivative of the fraction of total solar flux absorbed with respect to ozone column abundance from the parameterization of Lacis and Hansen (1974).
5.4 Routines used in the dynamical model

Subroutine HEAT computes the diabatic heating term for the dynamical equations. The earth-sun geometry factors and the set of Curtis matrices are computed once per day. The static stability is computed once, during the initial time step of the run. The zonal mean heating is introduced into the dynamical equations gradually with a growth time (GTIME) of two days.

Subroutine RADEQU computes the global radiative equilibrium temperature profile (i.e., the temperature profile such that the radiative heating integrated over the entire globe is zero at each level).

Function DELT computes the individual terms of the net radiative heating. The parameterization of Lacis and Hansen (1974) is used to compute the radiative heating due to absorption of solar visible and UV radiation by ozone. Values of the various parameters describing the shortwave planetary albedo are taken from the program of Ramanathan (private communication). The parameterization of Strobel (1978) is used for the absorption of solar radiation by molecular oxygen.

5.5 Changes in main program WAVE2

The flux equivalent width arrays are read in first for 10\(\mu\), then 15\(\mu\), in the order minus, zero, plus. KAP is the Newtonian cooling coefficient (day\(^{-1}\)) derived from the present algorithm, assuming a U.S. Standard temperature profile and the ozone profile (based on Krueger and Minzner, 1976) currently used in the dynamical model. The coefficients have been computed for a "boxcar" perturbation of vertical scale 25 km. TZO is on input the U.S. Standard atmosphere temperature profile. Array PRS contains the Fourier
decomposition of the observed $\bar{u}$ at the lower boundary. $Q_{O3}$ and $Q_0$ are $Q_3(\text{ground})$ and $Q_3$ of (2.1). $QOZSC$ and $QOZS$ are $n_3(\text{ground})$ and $n_3(p_k)$ of (2.2), $CTS$ is $-146.6c^g$ and $CG$ is $146.6c^g$, defined in (4.4).

Subroutine HEAT is called every IRCT time steps; IRAD is used as a counter.

Tropospheric and ground temperatures are computed with simple equations that fit the climatological temperatures to within a few degrees.
References


\begin{figure}[h]
\centering
\begin{align*}
\phi_1 &= 0 \\
\hat{\phi}_1, \hat{\theta}_1 \\
\phi &= \phi_2, \hat{\theta}_2 \\
\phi_3 &\quad \ddots \\
\phi_N &= 0.1 \\
\hat{\phi}_N &= \sqrt{0.1}, \hat{\theta}_N \\
\phi_{N+1} &= \hat{\phi}_{N+1} = 1, \hat{\theta}_{N+1}
\end{align*}
\caption{Vertical structure of atmospheric model.}
\end{figure}
Figure 2: Histograms of line strength distribution at three different temperatures for subinterval $650 \leq \nu \leq 675 \text{ cm}^{-1}$. Increments of $\log S$ are 0.25. Strength $S$ expressed in units of $\text{cm}^{-1} (\text{cm STP})^{-1}$. Lines under bracket are "strong lines", whose temperature dependence has been neglected.
Figure 3: Comparison of cool-to-space plus cool-to-ground approximation (broken line) with Curtis matrix method (solid line) for radiative cooling by the 9.6μ band of ozone. Temperature and ozone profiles are from the 1976 U.S. Standard Atmosphere.
Figure 4: Weak line distribution parameter $m$ vs temperature. Solid lines connect values for the same subinterval, identified on right (e.g., subinterval 1 is $550 \leq \nu \leq 575 \text{ cm}^{-1}$). Broken line is best fit for all subintervals.
APPENDIX

FORTRAN Code
PROGRAM BINSI (INPUT, OUTPUT, TAPE1, TAPE2, TAPE3, TAPE4, TAPE5, TAPE6, TAPE7, TAPE8, TAPE9
* TAPES 1 THROUGH 8 ARE ELEMENT FILES OF GROUP)
C SUBROUTINE TO SORT LINES ACCORDING TO STRENGTH
C (TAPES 1 THROUGH 8 ARE ELEMENT FILES OF GROUP)
C
DIMENSION N(100,7,50)
DIMENSION TP(7),QV(7)
INTEGER US
DATA TP/150,175,200,225,250,275,298,/
DATA QV/1.04,1.095,1.0142,1.0327,1.0502,1.0719,1.0931/

 ödül=2.0069E+19
C BEGIN OUTPUT SPRAYS
DO 11 IS=L,10
DO 9 IT=1,7
DO 3 I=1,20
N(IS,IT,1)=0
3 CONTINUE
9 CONTINUE
11 CONTINUE
C SORT EACH BAND
LD 12 LB=1,9
IS=1
UPLIN=579.
READ(18,15) LMAX
LL=0
99 LL=LL+1
IF(LL .GT. LMAX) GO TO 101
98 READ(10,5) FREQ,S,ALPHA,E,L,S,US,LS,ROT,DAT,ISO,MOLC
C IF(FREQ .LT. 5.0) GO TO 98
C IF(MOLC .NE. 2) GO TO 98
IF(FREQ .LT. UPLIN) GO TO 101
97 CONTINUE
GO 7 IT=1,7
1=IT
IF(EXP(1.43)*(LS*US/(T*TS))/IT)*IT)**2*FREQ)
C EXPRESS STRENGTH IN UNITS (1N ATM)-1 CM-1.
ST=ST+DSTP
SLT=ALOG10(ST)
C CHOOSE CORRECT BIN
IG=50
DO 31 IT=1,49
IF(SLT .GT. RL) IG=50-1.
31 CONTINUE
C COUNT LINES IN EACH BIN AND STOKE PARAMETERS
N(IS,IT,1)=N(IS,IT,IG)+1
7 CONTINUE
GO TO 99
101 CONTINUE
IF(IS .LT. 8) GO TO 13
OL=UPLIN-25.
WRITE(9,35) UL,UPLIN
DO 8 IT=1,7
WRITE(9,45) TP(IT)
DO 4 I=1,20
NN=N(IS,IT,1)
4 CONTINUE
WRITE(9,25) I,NN
13 UPLIN=UPLIN+25.
IS=IS+1
IF(UPLIN .GT. 800.) GO TO 12
GO TO 97
12 CONTINUE
23 FORMAT(10X,21B4,E15.2)
13 FORMAT(10X,15)
33 FORMAT(*) SPECTRAL INTERVAL*,F5.0,*, Temp*,F5.0)
43 FORMAT(10X,* TEMPERATURE*,F5.0)
END
PROGRAM FIT(INPUT,OUTPUT,TAPE1,TAPE2)
C SUBROUTINE IN FIT HISTOGRAMS WITH PARABOLAS
C (TAPE1 IS OUTPUT OF PROGRAM BINS)
C
DIMENSION N(51),LC(10)
DATA LC/16,9,8,3,0,1,5,9,11,10/
C LC IS BIN NUMBER OF WEAKEST EMPTY BIN IN EACH SUBINTERVAL
DO 4 I=1,10
READ(1,45) NUM
GO TO 17
READ(1,9) T
P=1.
TOT=0.
DO 1 I=1,51
N(I)=0
1 CONTINUE
D0 2 I=1,51
READ(1,15) N(I)
IF(N(I).EQ.0) GO TO 6
T=1e-14
TOT+1=TN
IF(TN.LT.P) GO TO 2
NC+1
P=TN
2 CONTINUE
IF(NC.GT.40) GO TO 3
IF(NC.GT.40) NC=40
IC=LC(15)
NP=NC-IC+1
S1=S2=0.
D0 7 I=1,MP
S1=S1+4*(IC+I-1)*(IC+I-1)*S1
S2=S2*(IC+I-1)**4
7 CONTINUE
RM=S1/S2
WRITE(2,25) IS,LC(IS),T,NC,TOT,NC
3 CONTINUE
WRITE(2,35)
4 CONTINUE
5 FORMAT(23X,F5.0)
25 FORMAT(2I5,F10.1,2F10.5,E10.4)
35 FORMAT(/)
45 FORMAT(10A6)
15 FORMAT(10X,216,E15.5)
END
PROGRAM REGRESS(INPUT,OUTPUT,TAPE1)
C SUBROUTINE TO COMPUTE BEST VALUES FOR UNIVERSAL FIT
C (TAPE1 IS OUTPUT OF PROGRAM FIT)

REAL M,N
SM=ST2+SM)-ST=0.
N=O.
UD 2 I=1,10
DD 3 I=1,7
READ(1,5) T,M
PRINT 5, T/M
M=M+1.
ST=ST+T
SMT=SMT+M*T
SM=SM+M
ST2=ST2+T*T
3 CONTINUE
READ(1,15) DUM
READ(1,15) DUM
2 CONTINUE
D=N*ST2-ST*T
A=(N*SMT-S*M*T)/0.
B=(SM-ST2-SMT+ST)/0.
PRINT 25, A,B
9 FORMAT(15X,F10.1,15X,F10.5)
15 FORMAT(10A6)
25 FORMAT(5A10,E15.5,10X,*8=9,E15.5)
END
PROGRAM RES10(INPUT, OUTPUT, TAPE1)
SUBROUTINE TO COMPUTE RESIDUAL STRONG LINES IN 15 MICRON BAND
(TAPE1 IS OUTPUT OF PROGRAM BINS)

DIMENSION N(10,7,91), XZ(10), A(7), SK(7), SS(7), R(I)
DATA XZ/16 9, 8, 3, 0, 1, 5, 9, 11, 16, 1/
DO 4 IS = 1, 10
READ(I,65) DIM
GO TO 3
READ(I, T) DO 1 I = 1, 51
N(I, I+1) = 0
1 CONTINUE
A(I) = 0.001989*T - 0.207
DO 2 I = 1, 51
READ(I, 10) N(I, I+1)
2 CONTINUE
A(I) = 0.001989*T - 0.207
GO TO 3
GO TO 3
CONTINUE
4 CONTINUE
READ(I, 65) DIM
PRINT 29
C RESIDUES FOR STRONGEST DECADE OF LINE STRENGTHS
DO 14 IT = 1, 7
KK = KK + A(I)*N(I)
SS(IT) = 0
R(IT) = 0.
IF(XZ(I) .LE. XZ(IT)) GO TO 16
XZ(IT) = XZ(IT) - XZ(I)
16 CONTINUE
R(IT) = R(IT) + A(I)*11. - XZ(IT) + 2
14 CONTINUE
PRINT 35, (SR(IT), IT = 1, 7)
PRINT 45, (SS(IT), IT = 1, 7)
C RESIDUES FOR SEVEN MORE DECADES OF LINE STRENGTHS
DO 9 K = 1, 7
DO 10 J = 1, 7
R(J) = R(J) + A(I)*N(I)
10 CONTINUE
DO 11 IT = 1, 7
KK = KK + A(I)*N(I)
R(IT) = R(IT) + A(I)*N(I) - XZ(IT) + 2
11 CONTINUE
PRINT 25, (K, RR, IT = 1, 7)
8 CONTINUE
DO 17 IT = 1, 7
IF(SS(IT) .EQ. 0.) GO TO 17
SS(IT) = SS(IT) + SR(IT)
17 CONTINUE
C MEAN FOR FOUR BINS (ONE DECADE)
PRINT 35, (SR(IT), IT = 1, 7)
PRINT 45, (SS(IT), IT = 1, 7)
9 CONTINUE
6 CONTINUE
25 FORMAT(15, 7F10.1)
35 FORMAT(7F10.1)
45 FORMAT(7E15.3)
15 FORMAT(10X, 2I6, E15.5)
55 FORMAT(1H1)
5 FORMAT(23X, F5.0)
65 FORMAT(10A6)
END
PROGRAM CURTEL(INPUT,OUTPUT,TAPE6,TAPE7)
COMMON/S(10,0),F(10,10),DELNU,ALPHDZ,E1(41,40),E(41,40)
* PHI(40),PHIH(40),TCG(20,20),TT(21)
DIMENSION T(21),R(41,40),Q(20),CTS(20),THETA(20)
DIMENSION H(21,20),U(20,20),W(20),U(20,20),W(20),R(20,21),Q(20)
DIMENSION F(10),S(10)
REAL NUZ,KB,MASS
DATA N,N,M1,L,DELTAH/174,163,2500.
DATA F/2.0,10.0,2.0,11.0,4.0/
DATA S/2.0,0.05,2.0,0.5,4.2,4.00/.
DATA T/47.1,267.4,193.2,210.6,221.0,724.0,234.7,248.4,283.9,47.00,725.0,1243.2,224.6,243.6,214.6,281.4,280.2,300/.
P=3.1415926
S1=SGN(T(21))
HP=6.6237E-27
AVG=6.0225E+23
MASS=44./AVG
C=2.979E+10
KB=1.3805E-16
CP=1.3E0
GRAV=981.
TJ=300.
C PARAMETERS FOR 15 MICRON BAND
NUZ=667.379
D=EXP(HP*CNuz/(KB*TJ))-1.
SB4=2.*HP*(C*NUZ)**3/(CP)
CUNST=2.*PI**3*GRAV/CP
ALPHDZ=NUZ*SQRT(2.*KB/TJ/MASS)/C
DELNU=DELTAH
C
DO 1 I=1,N
F(I,1)=FS(I)
S(I,1)=SS(I)/ALPHDZ
DO 2 J=2,N
F(I,J)=S(I,J)*0.
2 CONTINUE
DO 27 CONTINUE
C PRINT 101 N,M1,L,DELNU
CALL FLUXEL(N,M1,L)
DO 13 J=1,N
13 CONTINUE
WRITE(6,103)((F(I,J),S(I,J)),J=1,N)
13 CONTINUE
C COMPUTE LTE CURTIS MATRIX
DO 20 I=1,N
CTS(I)=0.
DO 21 J=1,N
R(J,I)=E1(J+1,I+1)=E1(J,I)=E1(J+1,I)=E1(J,I+1)
K(J,I)=ICONST/R(J,I)*R(J,I)*0.864000
CTS(I)=CTS(I)+R(J,I)
21 CONTINUE
20 CONTINUE
C WRITE(6,100)((R(J,I),J=1,N1),I=1,N)
WRITE(6,100) (T(I),I=1,N)
WRITE(6,102) WRITE(6,100) (CTS(I),I=1,N)
WRITE(6,102)
C COMPUTE TEMPERATURE THETA = B/8(B(R))
VZ=HP*CNuz/(KB*TJ)
DO 16 J=1,N
V=HP*CNuz/(KB*TJ)
THETA(J)=EXP(VZ-1.)/(EXP(V)-1.)
16 CONTINUE

C
C COMPUTE HEATING RATES
DO 10 I=1,N
Q(I)=0.
DO 9 J=1,N1
Q(I)=Q(I)+R(J,1)*THETA(J)
9 CONTINUE
WRITE(6,105) Q(I)
10 CONTINUE
C COMPUTE NON-LTE CURTIS MATRIX
CM=0.0003344/28.986
SBAND=0.
C COMPUTE TOTAL BAND STRENGTH FROM DISTRIBUTION
DO 23 I=1,10
DO 24 J=1,10
SBAND=SBAND+F(I,J)*S(I,J)
24 CONTINUE
23 CONTINUE
SBAND=SBAND*22414./44.
C PEKURM MATRIX ALGEBRA
DO 6 J=1,N1
DO 7 K=1,N
H(J,K)=0.
C EPS = COLLISION RATE/SPONTANEOUS EMISSION RATE
EPS=0.74*PHI(H(1),1,SC-05)
1IF(J,EQ.,K) H(J,K)=CP/(4.*PI*CM*SBAND*EPS*8BAR*86400.)
7 CONTINUE
6 CONTINUE
CALL VMULFR(P,H,N,N41,21,UB,20,IER)
PRINT 101, IER
DO 8 I=1,N
DO 11 J=1,N
U(I,J)=U(I,J)+1.
11 CONTINUE
8 CONTINUE
CALL LINV1F(U,N,20,UL,0,WK,IER)
PRINT 101, IER
DO 12 I=1,N
DO 13 J=1,N
SUM=SUM+UI(I,J)*R(K,J)
13 CONTINUE
12 CONTINUE
RR(K,1)=SUM
17 CONTINUE
18 CONTINUE
19 CONTINUE
WRITE(6,102)
C NON-LTE CURTIS MATRIX
WRITE(6,100) (RR(J,1),J=1,N1),I=1,N)
WRITE(7,100) (RR(J,1),J=1,N1),I=1,N)
C COMPUTE HEATING RATE FOR NON-LTE CURTIS MATRIX
DO 3 I=1,N
QG(1)=0.
DO 4 J=1,N1
QG(1)=QG(1)+R(J,1)*THETA(J)
4 CONTINUE
WRITE(6,105) QG(I)
3 CONTINUE
DO 12 I=1,N
CTS(I)=0.
DO 15 J=1,N1
CTS(I)=CTS(I)+RR(J,1)
15 CONTINUE
12 CONTINUE
WRITE(6,100) (CTS(I),I=1,N)
100 FORMAT(10E12.4)
101 FORMAT(13,F6.1)
STOP
END
SUBROUTINE FLUXEL(N,M,M1,L)
SUBROUTINE FOR CALCULATING FLUX EQUIVALENT BAND WIDTHS
COMMON/F10,10,F10,10,DEL4U,ALPHOZ,E(41,40),E(41,40)
*PHI(40),PHIM(40),TCG(20,20),T(21)
DIMENSION C(41,40)
N1=N+1
N2=M+2
DEL=DELNU/2.
IF(L.EQ.2) DEL=.49.
C
GO 10 I=1,N
PHI(I)=PFUNC(I,N)
10 PHI(M1)=1.
PHIM(N)=SORT(0.1)
PHIM(1)=1.
C
COMPUTE CURTIS-GOODSON TEMPERATURE FOR EACH PAIR OF LEVELS
DO 22 I=1,N1
DO 23 J=1,M1
TCG(I,J)=T(I)
IF(J.LT.0) J GO TO 23
SUM=0.
JL=J-1
DO 26 L=1,JM1
SUM=SUM+T(L)*T(L+1)*PHIM(L+1)-PHIM(L))
CONTINUE
TCG(I,J)=0.5*SUM/(PHIM(J)-PHIM(I))
TG(I,J)=TCG(I,J)
23 CONTINUE
PRINT 101,T(I),TCG(I,J),J=1,N1
22 CONTINUE
C
DO 70 I=1,N1
DQ 50 J=1,N
IF (J.EQ.1) GO TO 46
IF (J.EQ.I) GO TO 30
IF (J.GE.N) GO TO 48
C
E(J,1)= EFUNC(PHI(J),PHI(I),L)
COR=0.
C
COMPUTE NONDIAGONAL TERM
DO 15 K=1,M
EK=K-1
EK=K
PHIK=((M-EK1)/M)*PHIM(J-1)+((EK1/M)*PHIM(J)
PHIK1=((M-EK)/M)*PHIM(J-1)+((EK/M)*PHIM(J)
15 COR= COR+GFUNC(PHIK,PHIM(J),PHIM(J-1))*GFUNC(PHIK1,PHIM(J)
1,PHIM(J-1)))*EFUNC(PHIK1,PHI(I),L)-EFUNC(PHIK,PHI(I),L))
C
(J,J,1)=EFUNC(PHI(J),PHI(I),L)-E(J,1)-COR/2.
GO TO 50
C
C
COMPUTE DIAGONAL TERM
30 CORDN=0.
CORUP=0.
E(J,J)=0.
C
DO 35 K=1,M1
EK=K-1
EK=K
PHIK=((M-EK1)/M1)*PHMI(I-1)+((EK1/M1)*PHI(I)
PHIK1=((M-EK)/M1)*PHIM(I-1)+((EK/M1)*PHI(I)
35 CORDN= CORDN+GFUNC(PHIK,PHIM(I),PHIM(I-1))*GFUNC(PHIK1,
1PHIM(I-1)))*EFUNC(PHIK1,PHI(I),L)-EFUNC(PHIK,PHI(I),L))
C
(J,J)=EFUNC(PHI(J),PHI(I),L)+E(J,1)+COR/2.
C
GO TO 50
C compute non-diagonal term at top
46  C(I,J) = 0.
   E(I,J) = EFUNC(O., PHI(I), L)
GO TO 50

C
48  E(N,1) = EFUNC(PHI(N), PHI(I), L)
   E(N1,1) = EFUNC(PHI(N1), PHI(I), L)
   COR = 0.
   M2 = 3*N/2
   DO 49 K=1, M2
      EK = K-1
      PHIK = ((M2-EK)/M2)*PHI(N-1) + (E(K)/M2)*PHI(N1)
      PHIK1 = ((M2-EK)/M2)*PHI(N-1) + (E(K)/M2)*PHI(N1)
   49  COR = COR + (EFUNC(PHI, PHI(N1), PHI(N-1)) + EFUNC(PHI1, PHI(N1)))/(EFUNC(PHI1, PHI(N1)) - EFUNC(PHI, PHI(N1)))
   E(N1,1) = E(N1,1) - E(N,1) - COR/2.
C
50  E(J,1) = E(J,1) + C(J,1)
C
57  E(N1,1) = EFUNC(PHI(N1), PHI(I), L)
70  E(N2,1) = DEL
C
RETURN
END

FUNCTION PFUNC(J,N)
   RJJ = J - N
   PFUNC = 0.1*EXP(RJJ**5./7.)
   IF(J.EQ.1) PFUNC = 0.
RETURN
END

FUNCTION PHFUNC(J,N)
C THIS SUBROUTINE COMPUTES PRESSURE HALF LEVELS
   PHFUNC = SURT(PFUNC(J,N), PFUNC(J+1,N))
   IF(J.EQ.1) PHFUNC = 0.5*PFUNC(2,N)
RETURN
C
END

FUNCTION GFUNC(A,B,C)
C PLANCK FUNCTION LINEARLY INTERPOLATED IN LOG PRESSURE.
   GFUNC = ALOG(A/C)/ALOG(B/C)
RETURN
END
FUNCTION EFUNC(A,B,L)
COMMON/S(I0,10),F(I0,10),DELNU,ALPHDZ,B1(41,40),E(41,40)
S,PHI(40),PHII(40),TCG(20,20),T(T)
DIMENSION Y(Z)
DATA Y/0,2113./76.87/ 
RLFIVX=7,75=ALOG(100.*X)+17. 
IF(L .EQ. 1) GO TO 10 
IF(L .EQ. 2) GO TO 20 
IF(L .EQ. 3) GO TO 30
C
POLLACK 15 MICRON BAND,SLP=1000MB,C=0,00033
10 EFUNC=.055*ALOG(1.1336*((9.6543*AUL(A-B))**0.566)) 
1*1*(A+B)**2/2)
GO TO 60
C
RODGERS-WALSHAW 15 MICRON BAND,SLP=1000MB,C=0,0033 BY VOLUME
C
EVALUATED AT 200K
20 EFUNC=0. 
U=A**A(A-B) 
IF(U .EQ. 0.) GO TO 80 
P=(A+B)/2. 
TRAN=0. 
DO 21 J=1,2 
UP=U/Y(J) 
TRAN=TRAN+0.5*Y(J)*EXP(-300.3*UP/SQRT(1.+541.3*UP/P)) 
21 CONTINUE 
EFUNC=170.4*(0.5—TRAN) 
GO TO 90
C
DOPPLER-LORENTZ LINE
30 GRAV=981. 
EFUNC=0. 
C
COMPUTE PATH LENGTH 
U=(B-A)*1.0136*46.0.0.00033*22414./(GRAV*28.966) 
U=ABS(U) 
IF (U .EQ. 0.) GO TO 90 
T2=300.
C
CURTIS-GOWSON APPROXIMATION
IA=1 
IF(A .EQ. 1.) GO TO 2 
IA=RLEV(A) 
EP=RLEV(A)-IA 
1 IF(EP .GE. 0.5) IA=IA+1 
2 IB=1, 
IF(B .EQ. 0.) GO TO 3 
IB=RLEV(B) 
EP=RLEV(B)-IB 
1 IF(EP .GE. 0.5) IB=IB+1 
3 IF(IA > IB) 1A-IB 
IF(IB > IB) 18-IB+1 
T=TCG(IA,IB)
C
COMPUTE WEAK LINE DISTRIBUTION PARAMETER 
SL=0.601615 
BINT=-0.2066 
AZ=SL*T:PINT 
P=(A+B)/2. 
PZ=1. 
AL=0.08/ALPHDZ*P/PZ*TZ/T 
C
TWO POINT GAUSSIAN QUADRATURE OVER ANGLES 
TRAN=0. 
DO 22 J=1,2 
TRAN=TRAN+0.5*Y(J)*W(U/Y(J),AL,AZ) 
22 CONTINUE 
EFUNC=DELNU*(0.5—TRAN)
C
80 CONTINUE 
RETURN 
END
FUNCTION WIU,PA2)
C THIS SUBROUTINE COMPUTES MEAN TRANSMISSION FUNCTION
COMMON//S(10,P10),F(1U,10),DELNU,ALPHDZ,E(41,40),E(41,40)
DIMENSION XZ(10),N(10)
DATA X/16.,9.,8.,3.,0.,1.,5.,9.,11.,16./*
DATA N/883.,1407.,1676.,2640.,3724.,2660.,1916.,1770.,734.,493/
C M IS TOTAL NUMBER OF LINES IN EACH SUBINTERVAL
C XZ IS ONE LESS THAN NUMBER OF FIRST OCCUPIED BIN
PI=3.1415926
SPI=SORT(PI)
W=0.
DN=DELNU/10.
C TEN SPECTRAL INTERVALS
DO 0 I=1,10
XM=XZ(I)+1.*N(I)/A21++(1./3.)
DO 8 J=2,10
9=J-4
IFR.GT.U1 GO TO 5
IFR.LT.UI GO TO 6
VU=4.+R*9.5
IF(XM.LT.VU) VU=XM
VL=4.+R*5.5
IF(XZ(I).GT.VL) VL=XZ(I)
CONTINUE
5 CONTINUE
6 CONTINUE
C COMPUTE NUMBER OF LINES IN STRENGTH DECADE
F(I,J)=2./3.*{(VU-XZ(I))**3-(VL-XZ(I))**3}
C ESTIMATE MEAN STRENGTH IN DECADE
XMN=0.25*A20*(VU-XZ(I))**4-(VL-XZ(I))**4/F(I,J)*XZ(I)
S(I,J)=10**(-0.25*(XMN-5.5))
S(I,J)=S(I,J)/ALPHDZ
CONTINUE
B CONTINUE
C SUM OVER LINE STRENGTH DECADES
DO 4 J=1,10
S(U)=S(U)+S(I,J)
CONTINUE
4 CONTINUE
C GOLJMAN APPROXIMATION FOR LORENZ LINE EQUIVALENT WIDTH
WL=SU/(1.+((0.25*SU/A)**1.25)**0.4
IFSU.GT.4.1 GO TO 2
C SERIES APPROXIMATION FOR DOPPLER LINE EQUIVALENT WIDTH
WD=SU/SPI
GO TO 3
C LARGE SU APPROXIMATION FOR DOPPLER LINE EQUIVALENT WIDTH
U2=SQR((ALOG(SU/SPI))
WD=2.*((U2+0.2696/U2-0.1335/U2**3+0.0070)/U2**5)
C INTERPOLATION FOR VOIGT LINE
3 VV=SORT(WL,WD,VL,GL,WL,WD,/(SU*SU))
VV=VV*F(I,J)*VV
4 CONTINUE
W=EXP(-W*ALPHDZ/DELNU)
W=W*W*DN
CONTINUE
W=DELNU
RETURN
END
FUNCTION WD1(W)
C THIS SUBROUTINE COMPUTES DOPPLER LINE EQUIVALENT WIDTH FROM SERIES
PI=3.14159265
SPI=SQRT(PI)
NF=1
SUM=0.
DO 2 N=1,6
RN=N
NF=NF*N
R=NF
SUM=SUM+(W)**N/(R*SQRT(RN))
2 CONTINUE
WD1=SPI*SUM
IF(WD1 .LE. 0.) WD1=1.
RETURN
END
PROGRAM CUWD(INPUT,OUTPUT,TAPES,TAPES)
C THIS SUBROUTINE COMPUTE EQUIVALENT BAND WIDTH FROM CURTIS MATRICES
DIMENSION K(10,17),PHI(1),S(1,10),CST(2),ON(2)
REAL KB,NUZ(2)
PI = 3.1415926
HF = 0.6237e-17
C = 2.4974e+10
K& = 1.3058e-16
CP = 1.0e-17
GRAV = 9.81
T2 = 300.
DNZ(2) = 200. & ON(1) = 250.
NUZ(2) = 66.379
NUZ(1) = 979.
C FOR 10 AND 15 MICRON BANDS
DO 1 M = 1,2
U*EXP(HF*PHI(N1)/(KB*T2)) - 1.
BBE = 2. + HF*(C+NUZ(1))**3/(CO)
CUNST = 2.*PI*BBE + GRAV/CP
CST(1) = CUNST
1 CONTINUE
N = 17 & N1 = N1 + 1 & N2 = N2 + 2 & NM = N - 1
C CREAT VERTICAL GRID
DO 10 I = 1, N
PHI(I) = PFUNC(I, N)
10 CONTINUE
C CALCULATE FUNCTION E FOR SIX CURTIS MATRICES
C K(I,J),R0(I,J),R(I,J),R-1(J),RO(I,J),R+1(J)
DO 2 L = 1,3
CUNST = CST(2) & DELNU = UN(1)
IF(L1 & LT. 4) GOTO 3
CUNST = CST(2) & DELNU = UN(2)
2 CONTINUE
READ(5,25) (K(I,J), J = 1, N1), I = 1, N
DO 20 I = 1, N
OP = 1.013e+06*(PHI(I+1) - PHI(I))
DO 21 J = 1, N1
R(I,J) = OP*PHI(I,J) / (CUNST*0.400)
21 CONTINUE
20 CONTINUE
DU 9 I = 1, N1
E(I,1) = 0.
9 CONTINUE
C
DO 4 I = 1, N1
C(N2, I) = DELNU/2.
4 CONTINUE
C
DO 7 J2 = 1, N1
J = N2 - J2
JN = J - 1
DO 9 I = 1, JN
K = I - 1
E(I,1) = K(G(J,1) - E(J+1,1) + E(J+1,1) + E(J+1,1))
7 CONTINUE
6 CONTINUE
DU 12 K = 1, N1
K = K + 1
NK = N - K
DO 12 L = 1, NK
I = N - L
J = L
E(J,1) = R(J,1) + E(J+1,1) + E(J+1,1) - E(J+1,1)
12 CONTINUE
11 CONTINUE
WEIGHT(6.5) ((E(J,1), J = 1, N2), I = 1, N1)
DO 33 I = 1, N1
OP = 1.013e+06*(PHI(I+1) - PHI(I))
LU = J = J + 1
R(I,J) = OP*PHI(I,J) / (CUNST*0.400)
33 CONTINUE
31 CONTINUE
30 CONTINUE
C PRINT A T3 CHECK PROGRAM
PRINT 25, ((R(J,1), J=1,10), (N='1,17)
2 CONTINUE
5 FORMAT('I5E20.13/1,4E20.13)
25 FORMAT('I9F6.3)
EN0
SUBROUTINE CURT (TST, KN, TSTK)

SUBROUTINE TO COMPUTE TEMPERATURE-DEPENDENT CURTIS MATRICES

IT IS THE ACTUAL TEMPERATURE PROFILE, TST IS THE STANDARD TEMPERATURE

COMMON/CURTIS/PI (1D), PI (1D), GL (17), G2 (1D), E (6, 19, 10)

DECPMA (2D), DT (1D)

DIMENSION (1D), TS (1D), EL (2D), L (1D), TT (1D), TST (1D), R (2, 10, 17)

REAL K, NU (2)

K* = K - 1, KP = K + 1, A2 = K/2

CONVERT TO OLD VERTICAL COORDINATE CONVENTION

DO 2 K = 1, KNL

TS (K) = TST (K - 1)

T (K) = TST (K)

2 CONTINUE

TS (K) = TST (K)

T (K) = TST (K)

8 CONTINUE

COMPUTE TEMPERATURE DEVIATION FROM STANDARD

UU 1 = L, KP

OT (1) = T (1) - TST (1)

U (1, I) = DT (1)

B (1, I) = L

1 CONTINUE

COMPUTE WEIGHTED DEVIATION

UU 3 = L, KM

I P = I + 1

UU 4 = J, KP

UI (I, J) = OT (I) * GL (I) + DT (I) * G2 (J)

DI (I, J) = OT (I) * G2 (J)

IF (J .LT. KP) DI (I, J) = DI (I, J) + D1 (I) + J * (DT (KP) - DT (KM)) * GL (K)

L1 = I + 1

L2 = J - 1

L F (L1, L2) = 0 TO 7

DO 6 L = L1, L2

DI (I, J) = DI (I, J) + OT (L) * (UL (L) + G2 (L))

B (I, J) = B (I, J) + G1 (L) + G2 (L)

6 CONTINUE

7 CONTINUE

4 CONTINUE

3 CONTINUE

NORMALIZE DEVIATION

UU 9 = L, KM

I P = I + 1

UU 11 = J, KP

DI (I, J) = D (I, J) / B (I, J)

11 CONTINUE

9 CONTINUE

COMPUTE TEMPERATURE-DEPENDENT EQUIVALENT (BAND) WIDTHS

UU 12 = M + 1, X

UU 13 = L, KP

DO 14 J = 1, X2

E (I, J) = 3 * MJ (I, J) - (E (3 * MJ (I, J)) - E (3 * MJ (I, J) - 2)) * D (I, J) / 50.

14 CONTINUE

1 CONTINUE

12 CONTINUE

C OMPUTE CURTIS MATRICES

PI = 3.14159265

HP = 0.0237E-27

C = 2.907381E10

KM = 1.305056E-16

CP = 1.607

GRAV = 981

T2 = 300

NU (2) = 067.379

NU (1) = 949

UH = 1.12

UD = EXP (HP * C * NU (1) / (K3 * T2))

BBAR = 2.9 * HP * C * NU (1) * 3 / (C * UD)

CURST = 2. + PI * BBAR ^ 0.5

CURT
SUBROUTINE G(KN)
C SUBROUTINE TO COMPUTE WEIGHTING FUNCTION FOR TEMPERATURE-
C DEPENDENT CURTIS MATRIX CALCULATION
C COMMON/CURTIS/PHI(1:17),PHIM(1:18),U1(17),G2(16),E(6,19,18)
C X(K,35,17,16)
C G2(P1,P2)=5/6.*(P2**(6./5.)-P1**(6./5.))+(P2-P1**P1)/0.12098
C X(KN)=K-1 & X(K)-K-2
C CALL VERTICAL COORDINATE LABEL CONVENTION
C PHI(1)=0.,
C DO 1 K=2,KN
C PHI(K)=0.1*EXP(KJN*5./7.)
1 CONTINUE
C PHI(H(K+1))=PHI(K(K+1))=1.*
C PHI(K)=0.5*PHI(K)
C DO 2 K=2,KN
C PHI(K)=SUAT(PHI(K)*PHI(K+1))
2 CONTINUE
C DO 3 K=1,KN
C G2(K)=SUAT(PHIM(K),PHI(K+1))
C G2(K)=SUAT(PHIM(K),PHIM(K))
3 CONTINUE
C G2(KN+1)=0.
C RETURN
C END
PROGRAM QPRO
COMMON /E(41,40),E(41,40),
$PHI(40),PHIH(40),TCG(20,20),Q(18),QOS(18)
DIMENSION GUG(18),GG(18,17),QOZSG(18),QOZS(18,17),CTS(18,17),
$CTG(18,17)
DIMENSION W(18),CS(17),CG(17)
JN=19 	 KN=17
JML=JN-1 	 KNL=KN-1
KNP=KN+1
C COMPUTE OZONE PARAMETERS FOR EACH LATITUDE ZONE
DO 2 J=1,1
READ(1,3)(W(K),K=1,KNP)
READ(1,3)T(T(K),K=1,KNP)
C W IS MASS MIXING RATIO IN G/G AT PRESSURE HALF LEVELS
C NUMBERED DOWNWARD FROM TOP OF ATMOSPHERE. T IS TEMPERATURE AT SAME LEVELS
CALL GLO84(KN,W,TS,CG)
$UGT(J)=Q(KN)
SUZS(J,1)=QZS(KNP)
UG(J,KN)=Q(KN)
QOZS(J,KN)=QZS(KN)
CTS(J,KN)=CS(KN)
CG(J,KN)=CG(KN)
DO 3 K=1,KN
QO(J,K)=Q(KN-K)
SUZS(J,K)=QZS(KN-K)
CTS(J,K)=CS(KN-K)
CG(J,K)=CG(KN-K)
3 CONTINUE
2 CONTINUE
WRITE(2,195)(QUG(J),J=1,JML)
WRITE(2,195)(QUG(J,KN),J=1,JML)
WRITE(2,195)(QUG(J,K),J=1,JML,K=1,KNL)
WRITE(2,195)(QUGS(J,K),J=1,JML)
WRITE(2,195)(QUGS(J,KN),J=1,JML)
WRITE(2,195)(QUGS(J,K),J=1,JML,K=1,KNL)
WRITE(2,195)(QUGS(J,K),J=1,JML)
WRITE(2,195)(QUGS(J,KN),J=1,JML)
WRITE(2,195)(QUGS(J,KN),J=1,JML)
WRITE(2,195)(QUGS(J,KN),J=1,JML,K=1,KNL)
WRITE(2,195)(QUGS(J,KN),J=1,JML,K=1,KNL)
195 FORMAT(9E10.3)
5 FORMAT(E9.2)
1 FORMAT(F9.2)
END
SUBROUTINE GLOBE(KN,W,TS,CG)
DIMENSION W(18),CM(41,40),W(18),CTS(20),CG(20)
COMMON /E(41,40),E(41,40),
$PHI(40),PHIH(40),TCG(20,20),Q(10),QOZS(18)
KNP=KN+1
DO 70 I=1,KN
PHI(I)=PFUNC(I,KN)
70 CONTINUE
PHI(KNP)= 1.
DO 71 I=1,KN
PHIH(I)=PFUNC(I,PHI)
71 CONTINUE
PHIH(KNP)=1.
CALL COLUZ(KN,W,TS)
DO 72 L=1,KNP
PRINT 25, PHI(L),PHIH(L),W(L),QOZ(L),QOZS(L)
72 CONTINUE
CALL CMUZ96(KN,CM,CTS,CG)
29 FORMAT(5E15.5)
5 FORMAT(E9.2)
19 FORMAT(F9.2)
RETURN
END
SUBROUTINE COLO2(KN,W,T)
COMMON/E1(41,40),E4(41,40)
1,PHI(40),PHIH(40),TCG(20,20),QO(19),QOZS(18)
DIMENSION W(10),T(10)
REAL M3,MAIR
KNP=KN+1
MAIR=28.966
AVG=6.0223E+23
M3=44./AVG
K=9.3143E+07/MAIR
DSTP=2.6869E+19
ATMOS=1.013E+06
G=981.
C ASSUME OZONE SCALE HEIGHT IS H ABOVE TOP HALF LEVEL
H=4.7E-05
QO(1)=PHIH(1)*ATMOS*W(1)/(M3*DSTP*RA*T(1))
QOZ(1)=QO(1)*H
QO Z=Z+K/KNP
QO(K)=PHIH(K)*ATMOS*W(K)/(M3*DSTP*RA*T(K))
C ASSUME w VARIES LINEARLY WITH PRESSURE
QOZS(K)=QOZS(K-1)+0.5/(M3*DSTP*G)*(W(K-1)+W(K))*PHIH(K-1)*ATMOS
Z CONTINUE
RETURN
END

FUNCTION PFUNC(J,N)
C THIS SUBROUTINE COMPUTES PRESSURE LEVELS
RN=J-N
PFUNC=0.1*EXP(RN*5./7.)
IF(J .EQ. 1) PFUNC=0.
RETURN
END

FUNCTION PHFUNC(J,PHI)
C THIS SUBROUTINE COMPUTES PRESSURE HALF LEVELS
UP=PHIH(J)*PHIH(J+1)
PHFUNC=SQRT(UP)*PHI(J+1)
IF(J .EQ. 1) PHFUNC=0.5*PHI(2)
RETURN
END

FUNCTION GFUNC(A,B,C)
C FLAMCK FUNCTION LINEARLY INTERPOLATED IN LOG PRESSURE.
GFUNC= ALOG(A/C)/ALOG(B/C)
RETURN
END
SUBROUTINE CH0296(N,R,CS,CG)
  T=CH0296021,R(41,40),G(20),TS(20),THETA(20),CS(20),CG(20)
  R(20),CS(20),CG(20)
  COMMON/E1(41),E(41,40)
  PHI(40),PHIH(40),CG(20,20),G0(10),Q025(10)
  KSAL,NUZ,KB,MASS
  DATA M=1/1/1
  DATA T/187.1,186.9,193.2,201.6,210.4,221.0,234.7,248.4,263.9
  270.7,285.0,293.2,299.0,324.2,219.6,216.7,231.4,248.8,230.7/
C WrekBelh AMENDMENTS
  PRI=3.00000026
  SR=0.0000001
  HP=6.6257E-27
  M=1
  AVG=0.000000000000
  M9=4.4/AVG
  CG=2.079E+09
  K9=+1.00000000
  KG=+1.00000000
  CP=1.00000000
  GRAV=+9.81
  TZ=100.
C PARAMETERS FOR 9.6 MICRON BAND
  NUZ=1040.
  D=EXP((HP*CG*NUZ/(K9*TZ))-1).
  BBAX=+1.00000000
  C=1/(C*9)9/30+(C*8)
  CG=1.00000000
C COMPUTE CURTIS-GODSUN TEMPERATURE FOR EACH PAIR OF LEVELS
  DO 72 I=1,N1
    DO 73 J=1,N1
    TCG(I,J)=T(I)
    IF (1.EQ.J) GO TO 73
    SUM=0.
    J=J-1
    SUM=SUM+(T(I)+T(J))*PHI(I)-PHI(J))
    70 CONTINUE
    TCG(I,J)=0.5*SUM/(PHI(I)-PHI(J))
    TCG(I,J)=TCG(I,J)
    71 CONTINUE
    PRINT 100,(TCG(I,J),J=1,N1)
    PRINT 100,11R(J,1)
    PRINT 100,1T111,i+^1rN1)
    PRINT 102
    PRINT 102
    PRINT 102
    PRINT 102
C COMPUTE LTE CURTIS MATRIX
  DO 20 I=1,N
    CS(I)=0.
    UP=+1.013E+06*(PHI(I+1)-PHI(I))
    DO 21 J=1,N
      R(J)=C(I+1 J)+E1(J)+E1(J+1,1)-E1(J+1,1)-E1(J+1,1)
      R(J,I)=(COST/OP)*R(J)+1.06000.
      LS(I)+CS(I)*R(J,I)
      21 CONTINUE
    20 CONTINUE
C PRINT 100,1((R(J,I),J=1,N1),I=1,N)
    PRINT 102
    PRINT 100,1(T(I),I=1,N1)
    PRINT 102
    PRINT 101,1(R(I),I=1,N)
    PRINT 102
C COMPUTE COOLING RATES
  V=HP*C*NUZ/(KB*TZ)
  DO 10 J=1,N1
    V=HP*C*NUZ/(KB*TZ)
    THETA(J)=(EXP(VZ)-1.)/(EXP(V)-1.)
  10 CONTINUE
    PRINT 105,1(THETA(J),J=1,N1)
    PRINT 102
    DO I=1,N1
      CTS(I)=CS(I)*THETA(I)
      CG(I)=PH(I,I)
      1C(I)=CG(I)*(THETA(N1)-THETA(I))
SUBROUTINE FLUXEL(N,M,M1)
SUBROUTINE FOR CALCULATING CURTIS COEFFICIENTS FOR NET UPWARD FLUX
DIMENSION C(41,40)
COMMON/E1(41,40),E(41,40)
S,PHII(41),PHII(40),TCG(20,20),GO(18),QOL(S(18)
N1=N+1
N2=N+2

DO 70 I=1,N
DO 50 J=1,N
IF (J.EQ.1) GO TO 46
IF (J.EQ.N) GO TO 30
IF (J.EQ.M) GO TO 48

E(J,I)= EFUNC(PHII(J),PHII(I))
COR=0.

DO 15 K=1,M
EK*K=K
EK*K=K
PHIK= ((M-EK)/M)*PHII(J-1)+(EK/M)*PHII(J)
PHIK= ((M-EK)/M)*PHII(J-1)+(EK/M)*PHII(J)

CUR= CUR+GFUNC(PHII(J),PHII(J-1)+GFUNC(PHII,K,PHII(J)
1,PHII(J-1))*GFUNC(PHII,K,PHII(J))

C(J,I)= EFUNC(PHII(J),PHII(I)-E(J,I)-COR/2.
GO TO 51

C 30 COR=0.
C CUR=0.
C E(J,I)=0.

DO 39 K=1,M
EK*K=K
EK*K=K
PHIK= ((M-EK)/M)*PHII(I-1)+(EK/M)*PHII(I)
PHIK= ((M-EK)/M)*PHII(I-1)+(EK/M)*PHII(I)

CGD= CGD+GFUNC(PHII,K,PHII(I),PHII(I-1))+GFUNC(PHII,K,PHII(I)
1,PHII(I-1))*GFUNC(PHII,K,PHII(I))

PHIK= ((M-EK)/M)*PHII(I-1)+(EK/M)*PHII(I)

PHIK= ((M-EK)/M)*PHII(I-1)+(EK/M)*PHII(I)

35 CUR= CUR+GFUNC(PHII,K,PHII(I),PHII(I-1)+GFUNC(PHII,K
1,PHII(I-1)*PHII(I)*EFUNC(PHII,K,PHII(I)))

C C(J,I)= EFUNC(?PHII(1),PHII(I)+(CORN-CORUP)/2.

C GO TO 51
FUNCTION EFUNC(A,B)

C THIS SUBROUTINE COMPUTES FLUX EQUIVALENT WIDTH

C NOTE THAT NAME OF ABSORPTION FUNCTION APPEARS TWICE

DIMENSION Y(2)

COMMON/EL(I,40),E(41,40)

S ,PHI(40),PHIII(40),TCG(20,20),Q0(18),Q0Z5(18)

DATA Y/0.2113,0.7687/

EFUNC=0.

IF (A .GE. 0.1) GO TO 4

RETURN

4 CONTINUE

IF(U .EQ. 0. ) GO TO 80

RETURN

END

C END OF FUNCTION EFUNC(A,B)

46 C(I,J)-0.

E(I,J) = E FUNC(O ,PHI(I))

GO TO 51

C 49 E(N,J) = E FUNC(PHI(N),PHI(I))

L(N+1,J) = E FUNC(PHI(N1),PHI(I))

C0R = 0.

h2 = 3*M/2

DO 49 K = 1,M

EK = 1 - K

EK* = ( (H2-EK1)/H2) *PHII(N-I) + (EK1/M2)*PHI(N1)

PHI(K) = ( (H2-EK)/H2) *PHII(N-I) + (EK/M2)*PHI(N1)

C0R = COR10*EFUNC(PHI,K,PHI(N),PHI(N-1)) + G FUNCTION(PHI1,PHI(N))

I,PHI(N-I)) ) (EFUNC(PHI1,PHI(I))-EFUNC(PHI,K,PHI(I)))

C(N+1,J) = E(N+1,J)-E(N+1,J)-COR/K/2.

C 51 E(I,J) = E(I,J) + C(I,J)

50 CONTINUE

C E1(N1,J) = EFUNC(PHI(N1),PHI(I))

E1(N2,J) = E FUNC(-1. ,1.)

70 CONTINUE

C PRINT 102

PRINT 101, (E(J,1),J=1,8),E1(N1,1),E1(N2,1),I=1,N1

101 FORMAT(11F8.3)

PRINT 102

100 FORMAT(10E10.3)

102 FORMAT(1)

RETURN

END

FUNCTION EFUNC(A,B)

C THIS SUBROUTINE COMPUTES FLUX EQUIVALENT WIDTH

C NOTE THAT NAME OF ABSORPTION FUNCTION APPEARS TWICE

DIMENSION Y(2)

COMMON/EL(I,40),E(41,40)

S ,PHI(40),PHIII(40),TCG(20,20),Q0(18),Q0Z5(18)

DATA Y/0.2113,0.7687/

EFUNC=0.

IF(A .GE. 0.1) GO TO 4

RETURN

4 CONTINUE

IF(U .EQ. 0. ) GO TO 80

RETURN

END

C END OF FUNCTION EFUNC(A,B)

46 C(I,J)-0.

E(I,J) = E FUNC(O ,PHI(I))

GO TO 51

C 49 E(N,J) = E FUNC(PHI(N),PHI(I))

L(N+1,J) = E FUNC(PHI(N1),PHI(I))

C0R = 0.

h2 = 3*M/2

DO 49 K = 1,M

EK = 1 - K

EK* = ( (H2-EK1)/H2) *PHII(N-I) + (EK1/M2)*PHI(N1)

PHI(K) = ( (H2-EK)/H2) *PHII(N-I) + (EK/M2)*PHI(N1)

C0R = COR10*EFUNC(PHI,K,PHI(N),PHI(N-1)) + G FUNCTION(PHI1,PHI(N))

I,PHI(N-I)) ) (EFUNC(PHI1,PHI(I))-EFUNC(PHI,K,PHI(I)))

C(N+1,J) = E(N+1,J)-E(N+1,J)-COR/K/2.

C 51 E(I,J) = E(I,J) + C(I,J)

50 CONTINUE

C E1(N1,J) = EFUNC(PHI(N1),PHI(I))

E1(N2,J) = E FUNC(-1. ,1.)

70 CONTINUE

C PRINT 102

PRINT 101, (E(J,1),J=1,8),E1(N1,1),E1(N2,1),I=1,N1

101 FORMAT(11F8.3)

PRINT 102

100 FORMAT(10E10.3)

102 FORMAT(1)

RETURN

END

FUNCTION EFUNC(A,B)

C THIS SUBROUTINE COMPUTES FLUX EQUIVALENT WIDTH

C NOTE THAT NAME OF ABSORPTION FUNCTION APPEARS TWICE

DIMENSION Y(2)

COMMON/EL(I,40),E(41,40)

S ,PHI(40),PHIII(40),TCG(20,20),Q0(18),Q0Z5(18)

DATA Y/0.2113,0.7687/

EFUNC=0.

IF(A .GE. 0.1) GO TO 4

RETURN

4 CONTINUE

IF(U .EQ. 0. ) GO TO 80

RETURN

END

FUNCTION EFUNC(A,B)

C THIS SUBROUTINE COMPUTES FLUX EQUIVALENT WIDTH

C NOTE THAT NAME OF ABSORPTION FUNCTION APPEARS TWICE

DIMENSION Y(2)

COMMON/EL(I,40),E(41,40)

S ,PHI(40),PHIII(40),TCG(20,20),Q0(18),Q0Z5(18)

DATA Y/0.2113,0.7687/

EFUNC=0.

IF(A .GE. 0.1) GO TO 4

RETURN

4 CONTINUE

IF(U .EQ. 0. ) GO TO 80

RETURN

END

FUNCTION EFUNC(A,B)

C THIS SUBROUTINE COMPUTES FLUX EQUIVALENT WIDTH

C NOTE THAT NAME OF ABSORPTION FUNCTION APPEARS TWICE

DIMENSION Y(2)

COMMON/EL(I,40),E(41,40)

S ,PHI(40),PHIII(40),TCG(20,20),Q0(18),Q0Z5(18)

DATA Y/0.2113,0.7687/

EFUNC=0.

IF(A .GE. 0.1) GO TO 4

RETURN

4 CONTINUE

IF(U .EQ. 0. ) GO TO 80

RETURN

END

FUNCTION EFUNC(A,B)

C THIS SUBROUTINE COMPUTES FLUX EQUIVALENT WIDTH

C NOTE THAT NAME OF ABSORPTION FUNCTION APPEARS TWICE

DIMENSION Y(2)

COMMON/EL(I,40),E(41,40)

S ,PHI(40),PHIII(40),TCG(20,20),Q0(18),Q0Z5(18)

DATA Y/0.2113,0.7687/

EFUNC=0.

IF(A .GE. 0.1) GO TO 4

RETURN

4 CONTINUE

IF(U .EQ. 0. ) GO TO 80

RETURN

END
FUNCTION IU(X)
C THIS SUBROUTINE COMPUTES HALF LEVEL NUMBER JUST ABOVE PRESSURE X
IU=1
IF(X .LE. 0.) RETURN
IF(X .GT. 0.1) GO TO 2
IU=7./5.*ALOG10(10./1.4292*X)+17.
IF(IU .LT. 1) IU=1
RETURN
2 IU=16
IF(X .GT. 0.31623) IU=17
RETURN
END

FUNCTION IT(X)
C THIS SUBROUTINE COMPUTES HALF LEVEL NUMBER NEAREST PRESSURE X
C FOR CURTIS-GODSON TEMPERATURE APPROXIMATION
IT=1
IF(X .LE. 0.) RETURN
IF(X .GT. 0.1) GO TO 2
R=7./5.*ALOG10(10./1.4292*X)+17.
IT=R
E=P-1T
IF(E .GT. 0.5) IT=IT+1
IF(IT .LT. 1) IT=1
2 IU=17
IF(X .GT. 0.5623) IT=18
RETURN
END
FUNCTION OZ96A(U,T,P)
C THIS SUBROUTINE COMPUTES INTEGRATED ABSORPTION OF 9.05 MICRON
C OZONE BAND USING FORMULA OF AIDA (1975) JQSRT 15, PP. 389-403.
C IF(U .GE. 0.) GO TO 6
OZ96A=260.*U.
RETURN
6 CONTINUE
PI=3.141592654
SB=393.4
C CM-1 CMATM-1 AT 293.2 K
A=-0.1
DB=260.
X=SB*U/DB
G=0.076*P*(T/293.1)**(-0.7)
D=0.1
BT=2.*PI*G/0
Y=2.*X/BT
E=X/SQRT(1.+Y)
OZ96A=OR+((1.-EXP(-E)))
C Z=ALOG10((X/2.82)
C MODIFICATION TO FORMULA BY WMW
Z=ALOG10(P)
IF(U .GT. 2.) GO TO 1
A=0.16524*F**4+0.08623AF**3+0.54604AFAA2-0.11218AF-0.04140
A=10**A
B=(0.00412*F**3-0.05904*F**2+0.14168+F)*Z-0.04262*F**3-0.04262*F**2+0.50158+F-2.160
GO TO 2
1 A=-0.02097*F**3+0.11180+F**2-0.21613+F+0.21748
A=10**A
B=-0.4800*F**3+0.53*U
2 C=-0.01245*F**4-0.07222*F**3-0.10718+F**2+0.04269+F+0.55000
C CF=1.-C*EXP(-(Z**8)/A)
OZ96A=OZ96A*CF
RETURN
C ENTRY OZ96W
C THIS SUBROUTINE COMPUTES INTEGRATED ABSORPTION OF 9.6 MICRON
C OZONE BAND USING FORMULA OF WALSHAW (1957) JRMS 83, PP. 315-321
C IF(U .GE. 0.) GO TO 7
OZ96W=136.*U.
RETURN
7 CONTINUE
X=(U+0.1925*U)/(U+1.61+U)
IF(U .LE. 0.1) GO TO 3
X=0.934*10**(-U.534*U)
IF(U .LE. 0.4) GO TO 4
X=0.317*U**(-0.74)
A=2.11
G=U*X**A/(760.*P)
GL=ALOG10(G)
C TABULATED FUNCTION FITTED WITH STRAIGHT LINE SEGMENTS
E=1.
IF(GL .LT. -4.2) GO TO 3
E=-0.0176*(GL+3.8)+0.992
IF(GL .LT. -2.8) GO TO 3
E=0.1348*(GL+3.0)+0.948
3 F=1.165/SQRT(1.+734.*G)+E
OZ96A=138.*U-10**(U*X+F)
RETURN
END

ORIGINAL PAGE IS OF POOR QUALITY
SUBROUTINE OXYTPKN

SUBROUTINE TO COMPUTE DENSITY AND COLUMN ABUNDANCE OF MOLECULAR OXYGEN

ASSUME N(O2)=0.2095+N(10TA1) TO 155 KM

COMMON/O2/O2(17),CAZ(17)

DIMENSION T(17),REAL KM

KML=KM-1

K2=1.300*E-16

P=0.6996+EXP(RJN*5./7.)

IFJ (EQ, KM) P=0.5*EXP(RJN*5./7.)

IFJ (EQ, KM) P=0.5*EXP(RJN*5./7.)

DO2J+0.2095-.013F+06/(K2*1.0)

CAZJ=1.711E+15*4.496E+24*(P-3.695E-09)

2 CONTINUE
RETURN
END

FUNCTION O2UV(U)

SUBROUTINE TO CALCULATE SOLAR ENERGY ABSORBED BY MOLECULAR OXYGEN

PARAMETERIZATION BY STRABILL (1972) J. OCEPHYS RES 63 PP. 6229-6230

SCHUMANN-RNGE CONTINUUM, 1250-1320 A

LX1=0.

LX1=1.0E-17+U

IF(E1 .LT. -130.) GO TO 3

LX1=EXP(E1)

3 Q1=1.1E-177*1

SCHUMANN-RNGE CONTINUUM, 1520-1750 A

EX<0.

EX=1.54E-18+U

IF(EZ .LT. -100.) GO TO 4

EX=EXP(EZ)

4 EX=0.

EX=1.54E-18+U

IF(E4 .LT. -100.) GO TO 7

EX=EXP(E4)

7 QZ=(3.43*EX2-2.8*EXJ-1.35*EX4)/U

SCHUMANN-RNGE BANDS

QJ=0.036-14

IF(U .LT. 1.0E+13) GO TO 2

QJ=1.0/0.143(U+0.64)+.5*WRT(U)

2 O2UV=1+Q2+Q3
RETURN
END

FUNCTION O2UV(U)

SUBROUTINE TO CALCULATE SOLAR ENERGY ABSORBED BY OZONE

(DEVIATIVE OF LACIS-HANSEN EXPRESSION)

F1 = 1.0+(0.042+U)*1.23E-4*(U+2.01)

F2 = (0.0212/F1)+(1.0-((U/F1)+(0.042+(0.46E-4+U)))

F1 = 1.0+(138.6*U)

F2 = (F2*(1.082/F1+(0.805))*(1.0-((138.6+0.805))/F1))

F1 = 1.0+(138.6*U)

O2UV = F2*(0.0058/F1)+(1.0-(13.0*(F1-1.0)/F1))

RETURN
END

ORIGINAL PAGE IS OF POOR QUALITY
SUBROUTINE HeatTime, G0V, PB, TB, QB, KNH, DENS, OZ, JM, KN, T2O, BVF, SH, PI

DO 65 J = 1, JNL

DO 65 K = 1, KNL

65 TP(J,K) = TP(J,K-1) + OZ(J,K) / DZ

CONTINUE

DATA TS/216.719.6224.2229.6243.2258.0270.7263.9, 248.4,

34.7, 221.0, 210.6, 193.2, 186.9, 177.1, 231.4, 256.2/"}

JML = JM - 1

K ML = KN - 1

KNL = KN + 2

J2 = JML / 2

C COMPUTE ZONAL MEAN TEMPERATURE (DEVIATION FROM GLOBAL AVERAGE)

DO 66 K = 1, KNL

66 TP(K) = (TP(K) + TP(K-1)) / 2

CONTINUE

C CHOOSE APPROPRIATE TROPOSPHERIC TEMPERATURE

TP(K) = TROP(K)

C COMPUTE SOLAR GEOMETRY FACTORS

DAY = TIME / (24. * 60. * 60.)

DAYS = DAY - 0.05

RES = DAY - IDAY

EPS = 0.7 * DELTIM / (24. * 60. * 60.)

IF (EPS .GT. 0.05) GO TO 60

C COMPUTATION OF EARTH-SUN DISTANCE (BERGER, 1977)

EC = 0.016722

BETA = SQRT(1. - EC * EC)

OMB = (101.21972 + 20.0) * PI / 180.

KLAMB = -2. * (EC / 2. + (EC * 0.3) / 8. + (1. + BETA) * SIN(-OMB))

SL = (EC / 2. + (EC * 0.3) / 8. + (1. + BETA) * SIN(-OMB))

KLAMB = KLAMB + DAY / 360. * 2. * PI

RLAMB = RLAMB + DAY / 360. * 2. * PI

RLAMB = RLAMB + (2. * EC - 0.25 * EC * 3.) * SIN(RLAMB - OMB)

C SEASONAL VARIATION OF SOLAR HEATING (GOOGLEY AND BORUCKI, 1976)

DELTA = 0.4093196 * SIN(Z.*PI/(STROAY + 01.0/360.))

IF (DELTA .LT. 0.25) GO TO 25

TSTAR = 12.

LEN = SIN(DELTA) * COS(PHI) - TSTAR

ZEN = COS(DELTA) * COS(PHI) - TSTAR

IF (LEN .LT. 0.25) GO TO 25

CONTINUE
HEAT

AIZEN = 35./SORT(1224.*ZEN*ZEN+1.)
FDAY = TSTAR/12.
ZN(L) = ZEN
AZNL(L) = AIZEN
FOY(L) = FDAY
30 CONTINUE

C
C COMPUTE TEMPERATURE-CORRECTED CURTIS MATRICES FOR EACH ZONE

CALL CURT(TP,KN,T5,CM)

C CONVERT TO JRM VERTICAL COORDINATE LABEL

DO 69 JJ=1,KNP
   DO 69 II=1,KNL
   R(JJ+1,II)*=CM(1,KNZ-JJ,KN-1)
   R(JJ+1,II)*=CM(2,KNZ-JJ,KN-1)
69 CONTINUE

R(JJ+1,1)*=CM(1,KNZ-JJ,KN)
R(JJ+1,KNL)*=CM(2,KNZ-JJ,KN)
68 CONTINUE

66 CONTINUE

C COMPUTE GLOBAL RADIATIVE EQUILIBRIUM TEMPERATURE

CALL RADEQU(TZO,RHOPJ,KN)

IF (TIME.GT.0.) GO TO 76

C COMPUTE STATIC STABILITY FROM TZO

N = KNL-1
DO 70 K=2,N
70 UBV(K) = BVF+RHZ/(2./TZO(K)/SH+(TZO(K+1)-TZO(K-1))/(2.*DZ))
GO TO 73

73 CONTINUE

60 CONTINUE

C COMPUTE HEATING RATE

DO 40 J=1,KNP
40 CONTINUE

DO 45 L=1,KNL
45 CONTINUE

GTIME = 1.73E+06
DO 85 K=1,KNL
   DO 85 L=1,KNL
85 CONTINUE

90 CONTINUE
105 FORMAT (5X,8E15.3)
RETURN
END
SUBROUTINE RADQU(TZ0,RHO,JM,KN)
DIMENSION TZ0(17),A3(2)
DATA ILOT,DT,EPS/20,0.1.,0.1./
JML=JM-1 "$ KNL=KN-1"
PI=3.141592654
DO 40 IT=1,ILIT
SUM=0.
C IMAX IS KN FOR CLIMATOLOGICAL TROPOSPHERIC TEMPERATURE,
C KN FOR RADIATIVE EQUILIBRIUM TROPOSPHERIC TEMPERATURE
IMAX+KNL
DO 35 J=1,IMAX
TZ0(I)=TZ0(I)+2.*DT
DO 30 J=1,2
TZ0(I)=TZ0(I)+DT
A3(J)=0.
DO 25 L=1,JML
THETA=175.(L-1)*10.
THETA=THETA*PI/180.
W=SIN(THETA)*PI/30.
A3(J)=A3(J)+DEL(TZ0,L,1,RHO,KN)*W
25 CONTINUE
30 CONTINUE
DO = (A3(1)-A3(2))/DT
DIFF=A3(2)/DO
IN = TZ0(1)-DIFF
OTP = DIFF
IF (OTP.GT.20.) DIP = 20.
IF (OTP.LT.-20.) OTP = -20.
TN = TZ0(1)-OTP
DIFF=ABS(DIFF)
IF (DIFF.GT.SUM) SUM = DIFF
TZ0(1) = TN
35 CONTINUE
IF (SUM.LT.EPS) GO TO 45
40 CONTINUE
PRINT 65, ILIT
STOP 1
45 PRINT 70, IT
WRITE(6,75) TZ0(KN),(TZ0(K),K=1,KN)
RETURN
C
65 FORMAT (5X,*TEMPERATURE PROFILE FAILED TO CONVERGE AFTER*13,
* ITERATIONS*)
70 FORMAT (5X,*TEMPERATURE CONVERGED AFTER*12,* ITERATIONS*)
75 FORMAT (1X,18F7.2)
END
FUNCTION DELT(TP,L,I,RHO,KN) DELT
COMM/BUMLG/TG(C18),TRUP(L1),PRS(7,19),WOG DELT
S(18),QO(14,17),QJSS(16),QO2S(10,17),CS(18,17),CG(18,17),IN(18), DELT
AZN(18),FOY(18),CTZ(19,17),STZ(19,17),COT(19,17) DELT
COMM/BUMLG/TG(18),PHI(18),PHIW(18),G(18),GZ(18),E(6,19,18) DELT
R(14,17,18) DELT
COMM/WZUDZ(17),ZDZ(17) DELT
DIMENSION TP(17),TO(18) DELT
KMN-KN-1 & KMP=KN+1 DELT
KML=10 DELT
C 1.1. AMENDMENTS, 20 OCTOBER, 1980 C
V=HR*C=WN/KN DelT
V15=959.96 $ V10=409.64 DelT
C=C1G=U. DelT
C CHOOSE: /KUPRIAE TROPOSPHERIC TEMPERATURE DELT
TO(1)=TG(L) DELT
TO(2)=TRUP(L) DELT
TO(2)=TP(KN) DELT
UG 4 $3,KNP DelT
TD(J)=TP(J-2) DelT
4 CONTINUE DelT
D 3 J=1,KMP DelT
TT=TO(J) DelT
TH15=(EXP(V15/300.0)-1.)/(EXP(V15/TT)-1.) DelT
TH10=(EXP(V10/300.0)-1.)/(EXP(V10/TT)-1.) DelT
C15=AZN אליו+JML+1)*TH15 DelT
C10=AZN+R(L,J)*TH10 DelT
3 CONTINUE DelT
CO2=C15+C10 DelT
CUT(L,I)=CO2 DelT
CTS=C1S(L,I)*EXP(-1500./TP(I)) DelT
CTG=C1G(L,I)*EXP(-1500./TP(I)) DelT
COZONE=CTS*CTG DelT
CIT(L,I)=COZONE DelT
DELT=CCU2*COZONE DelT
SQ=0. DelT
STC(L,I)=50 DelT
ZEN=IN(L) DelT
AZN=AZN(L) DelT
FDAY=FOY(L) DelT
1F1D(DAY $LT. 0.0001) GO TO 25 DelT
DZ = 5000 DelT
SH = 7000 DelT
Z = (1-J-1)*DZ DelT
AIRDEN=1.033*Qo*0.00996*EXP(-Z/S)(2.87E+06*TP(I)) DelT
IF(I < QK, KN) AIRDEN=1.103*Qo*0.3162/(2.87E+06*TD(2)) DelT
TAU = 10 DelT
CLOUD = 0.446 DelT
RG = 0.9 DelT
ICLJUD=KN DelT
C HEATING BY THE ABSORPTION OF SOLAR RADIATION BY OZONE DelT
C FROM LACIS AND HANSEN, J ATMOS SCI 31 118-153 DelT
C CLEAR SKY DelT
RAB = 0.219/(1.+0.816*ZEN) DelT
KDB = 0.144 DelT
R81 = RAB*(1.-RAB)*(1.-RDB)*RG/(1.-RDB*RG) DelT
KB = R81 DelT
C CLOUDY SKY DelT
RAB = 0.13*TAU/(1.+0.13*TAU) DelT
ROB = RAB DelT
R81 = RAB*(1.-RAB)*(1.-RDB)*RG/(1.-RDB*RG) DelT
UT = AZN*QO2S(L1,CLOUD)+1.9*(QO2S(L1,CLOUD)-QO2S(L1)) DelT
A1 = OZUV(UT) DelT
UT = AZN*QO2S(L1)+1.4*(QO2S(L1)-QO2S(L1)) DelT
A2 = OZUV(UT) DelT
U = AZN*QO2S(L1) DelT
A3 = OZUV(U) DelT
CP=1.4*07 6 SOLFLX=0.1365E+07 DelT
SC3=FDAY*80400.*SOLFLX*ZEN*QOZ(1,L1)/(CP*AIRDEN)*(AZN*AZ) DelT
$CLOUD=1.9*R81*A1/(1.-CLOUD)*1.9*R8*A2 DelT

ORIGINAL PAGE 27 OF BOOK 27
U=ALEN*CA2(I)
A=O2UV(I)
502=A*O22(I)*85400.*FDAY/(AIRDEN*CP)
50=(SO3+SO2)/(KHO*KHO)
S2(TJ)=50
25 DELT+DELT+SO
RETURN
END
(Changes in main program)

```
READ(2,193)((F(M,J,I),J=1,KN2),I=1,KNP),M=1,6)
193 FORMAT(3(5E20.13),4E20.13)
READ(2,191)(KAP(I),I=1,KN)
READ(2,200) TZO(KN),TZO1(I),I=1,KNL)
191 FORMAT(F7.4)
READ (2,195) (PRS(I,J),I=1,4),J=1,19)
READ(2,195) QO5,(QO(L,K),L=1,JML),((QO(L,K),L=1,JML),K=1,KNL)
195 FORMAT(4F15.4)
145 FORMAT (9E10.3)
200 FORMAT (9F10.3)
```

```
DO 20 K=1,KN
   
   IF (K.GT.KNL) GO TO 20
   
C NEWTONIAN COOLING COEFFICIENTS FOR EDDIES
   KAP(K)=KAP(K)*DT/86400.
20 CONTINUE
   
IRAD = 0
IRCT = 12
```

```
   
   C COMPUTE ZONAL MEAN TEMPERATURE (DEVIATION FROM GLOBAL AVERAGE)
   DO 64 J=1,JML
      DO 64 K=1,KNL
      64 TB(J,K)=RH2*(PB(J,K)*DENS(K+1)-PB(J,K)*DENS(K))/DZ
      DU 66 J=1,JML
      DO 67 K=1,KNL
         TP(K) = TR(J,K) + TZO(K)
         67 CONTINUE
   
   C CHOOSE APPROPRIATE TROPOSPHERIC TEMPERATURE
   TP(KN) = TRUP(J)
   C TP(KN) = TZO(KN)
      TP(KNP) = TGE(J)
   C COMPUTE TEMPERATURE-CORRECTED CURTIS MATRICES FOR EACH ZONE
   CALL CURT(TP,KN,TZ,CMAT)
   C CONVERT TO JPH VERTICAL COORDINATE LABEL
   DO 68 JJ=1,KNP
      DO 69 II=1,KNL
         CURMAT(J,J,JJ) = CMAT(1,KN2-JJ,KN-JJ)
         CURMAT(J,JM,L,JJ) = CMAT(1,KN2-JJ,KN-JJ)
         69 CONTINUE
      DO 80 JJ=1,KNP
         CURMAT(J,KN,JJ) = CMAT(1,KN2-JJ,KN)
         CURMAT(J+JML,KN,JJ) = CMAT(1,KN2-JJ,KN)
         80 CONTINUE
   68 CONTINUE
   69 CONTINUE
   80 CONTINUE
   200 FORMAT (4F15.4)
   145 FORMAT (9E10.3)
```

C COMPUTE INITIAL TROPOSPHERIC AND GROUND TEMPERATURES

DO 96 L=1,J2
   PHD=L*1U-5
   TAMP=(0.475*PHD-4.)/2.
   ARG=2.*PI/360.,*(STROAY-186.)
   TEQX=54.03*(COS(PHD*PI/180.)*-U.22)-15.
   TG(J2+1-L)=TEQX+TAMP+CGS(ARG)
   TG(J2+1-L)=TG(J2+1-L)+273.16
   TG(L+J2)=TEQX-TAMP+COS(ARG)
   TG(L+J2)=TG(L+J2)+273.16
   TAMP=.3914*(PHD-0.)/2.
   IF(PHD .GT. 40.) TAMP=0.
   ARG=2.*PI/360.,*(STROAY-180.)
   TEQX=26.13*(COS(PHD*PI/180.)*-0.26)-54.
   IF(PHD .LT. 30.) TEQX=-32.
   TROP(J2+1-L)=TEQX+TAMP+COS(ARG)
   TROP(J2+1-L)=TROP(J2+1-L)+273.16
   TROP(J2+L)=TEQX+TAMP+COS(ARG)
   TROP(J2+L)=TROP(J2+L)+273.16

96 CONTINUE