Review and Improvement of Methods for Estimating Rates of Photolysis in Photochemical Models


Final Report
Contract No. 96-335

Prepared for:
California Air Resources Board and the California Environmental Protection Agency

Prepared by:
Laurent Vuilleumier and Nancy J. Brown
Ernest Orlando Lawrence Berkeley National Laboratory

Robert A. Harley
University of California at Berkeley

and

Steven D. Reynolds
Envair, 12 Palm Avenue, San Rafael, California, 94901-2222

December 8, 2000
Disclaimer

The statements and conclusions in this Report are those of the contractors and not necessarily those of the California Air Resources Board. The mention of commercial products, their source, or their use in connection with material reported herein is not to be construed as actual or implied endorsement of such products.
Acknowledgments

The authors thank James Pederson of the California Air Resources Board and the staff of CARB, Sasha Madronich of the National Center for Atmospheric Research, Linsey Marr of the University of California at Berkeley and James Slusser and David Bigelow from the Natural Resource Ecology Laboratory for their valued advice.

This report was submitted in fulfillment of ARB contract 96-335, Review and Improvement of Methods for Estimating Rates of Photolysis in Photochemical Models, by Lawrence Berkeley National Laboratory and the University of California at Berkeley under sponsorship of the California Air Resources Board. Work was completed as of May 31, 2000.
# Table of Contents

Disclaimer ....................................................................................................................... ii  
Acknowledgments .......................................................................................................... iii  
Table of Contents ........................................................................................................... iv  
1 Introduction .............................................................................................................. 1  
2 Theoretical background ............................................................................................. 2  

- 2.1 Actinic flux and photolysis rate coefficients .......................................................... 2  
- 2.2 Radiation transfer general equation ........................................................................ 3  
- 2.3 Two-stream methods ............................................................................................ 4  
- 2.4 Other methods ..................................................................................................... 8  
3 Using the library within an AQM .............................................................................. 8  
4 Using the library with a standalone program ........................................................... 10  
5 General card and command description ................................................................... 10  

- 5.1 Card description and syntax ................................................................................ 11  
  
  - 5.1.1 Card list ........................................................................................................... 13  
  
  - 5.1.1.1 General parameters: ...................................................................................... 13  
  
  - 5.1.1.2 Wavelength and elevation general grid choice: ............................................. 15  
  
  - 5.1.1.3 Air pressure (air density) definition: .............................................................. 16  
  
  - 5.1.1.4 Scaling and interpolation .............................................................................. 18  
  
  - 5.1.1.5 Aerosol definition: ....................................................................................... 19  
  
  - 5.1.1.6 Cloud definition: ......................................................................................... 22  
  
  - 5.1.1.7 Ground albedo definition: .......................................................................... 23  
  
  - 5.1.1.8 Atmospheric temperature definition: ............................................................ 24  
  
  - 5.1.1.9 Ozone elevation profile definition: ............................................................... 26  
  
  - 5.1.1.10 SO₂ molecule elevation profile definition: ................................................. 29  
  
  - 5.1.1.11 NO₂ molecule elevation profile definition: .................................................... 30  
  
  - 5.2 Command description and syntax .................................................................... 32  
  
  - 5.2.1 Command list .................................................................................................. 33  
  
  - 5.2.1.1 End-of-file command .................................................................................. 33  
  
  - 5.2.1.2 Sensitivity analysis SCAN command ......................................................... 33  
  
  - 5.2.1.3 Atmospheric light-absorbing gas cross-section definition ......................... 34  
  
  - 5.2.1.4 Extraterrestrial solar flux definition .............................................................. 37  
  
  - 5.2.1.5 Photolysis reaction cross section and quantum yield definition ............... 40  
  
6 Region and time-dependent card description ........................................................... 42  
7 Installing TUVAQM ................................................................................................ 48
8 Code description ................................................................................................................. 50
  8.1.1 Generic routines ............................................................................................................. 50
      8.1.1.1 tuvaqm.f ................................................................................................................. 50
      8.1.1.2 dep_rd.f ................................................................................................................. 50
      8.1.1.3 getpar.f ................................................................................................................. 51
      8.1.1.4 gtploc.f ................................................................................................................. 51
      8.1.1.5 modpar.f ................................................................................................................. 51
      8.1.1.6 tuv_rd.f ................................................................................................................. 51
      8.1.1.7 tuv_rn.f ................................................................................................................. 51
      8.1.1.8 zg_mod.f ................................................................................................................. 52
  8.1.2 Test programs ............................................................................................................... 52
      8.1.2.1 mn_try.f ................................................................................................................. 52
      8.1.2.2 mn_sca.f ................................................................................................................. 52
      8.1.2.3 tstaqm.f ................................................................................................................. 52
      8.1.2.4 tstaqmh.f ................................................................................................................. 52
  8.1.3 Other routines ............................................................................................................... 53
      8.1.3.1 setsca.f .................................................................................................................. 53
      8.1.3.2 caread.f .................................................................................................................. 53
      8.1.3.3 inidef.f .................................................................................................................. 53
      8.1.3.4 setcnd.f .................................................................................................................. 54
      8.1.3.5 setphr.f .................................................................................................................. 54
      8.1.3.6 setflx.f .................................................................................................................. 55
      8.1.3.7 setxsn.f .................................................................................................................. 55
      8.1.3.8 tuvgrd.f .................................................................................................................. 55
      8.1.3.9 rad_tr.f .................................................................................................................. 56
      8.1.3.10 tuvutl.f .................................................................................................................. 56
      8.1.3.11 outpaw.f ................................................................................................................. 57
      8.1.3.12 tuvold.f ................................................................................................................. 57

Literature cited .......................................................................................................................... 58
1 Introduction

This manual describes the TUVAQM module for calculating solar actinic flux and photolysis rates within air quality models (or on a standalone basis), based on the Tropospheric Ultraviolet Visible (TUV) radiation transfer program (version 3.9) by Dr. Sasha Madronich of the National Center for Atmospheric Research (NCAR). TUVAQM is designed to be callable from another program such as an Air Quality Model (AQM). The package includes library routines, test programs, UNIX installing scripts, and supporting data. The object of TUVAQM is to compute the solar actinic flux at a given location and time given specific atmospheric conditions. Photolysis reaction rate coefficients can be computed by TUVAQM providing the cross sections and quantum yields are available. The relationship between solar actinic flux and photolysis reaction rates, as well as the definition of these quantities is discussed in section 2. Section 2 also summarizes the theoretical background that supports the radiation transfer model.

The radiation transfer calculations depend on a large number of parameters that describe the atmospheric optical and other properties. Most of these parameters are given reasonable default values in the library routines, however the user can (and sometime must) provide some parameter values for describing specific situations. The parameter values can be set via input files whose syntax is described in section 5 and 6. The library routines can either be used within an Air Quality Model or for simulating a standalone radiation transfer model.

When TUVAQM is used within a standalone radiation transfer program (see section 4), the situation is unique for each simulation, corresponding to the conditions at a given time and location. Thus, the parameters values are fixed for the given time and location, and their values can be set with the input file described in section 5.

When TUVAQM is used within an air quality model (see section 3), a range of conditions can occur depending on the modeling domain cell (location) and modeling time. Some parameters may remain constant throughout the whole domain and modeling episode, and some others may vary from cell-to-cell and time-to-time. The approach chosen is to select a reduced number of important parameters that are given values that depend on time and space and let the other parameters apply to the whole AQM simulation. The parameters that depend on time and space can be set with the input file described in section 6. Parameters that do not vary spatially or temporally are set in the same manner as for a standalone simulation (file described in section 5).

TUVAQM has been installed in the following air quality models: Urban Airshed Model (UAM-IV), Urban Airshed Model with flexible chemical mechanism (UAM-FCM), and the SARMAP Air Quality Model (SAQM). For use of TUVAQM with these air quality models, the user should
review sections 3 and 4, and the execution instructions in sections 5 and 6. If the user wishes to install TUVAQM in another air quality model or install a standalone version of the program, a more detailed knowledge of the module structure is needed, and the user should also review installation instructions (section 7) and the code description (section 8).

2 Theoretical background

This section summarizes the key elements for establishing the mathematical expressions used to represent the radiative transfer in TUVAQM. A more detailed description is given in section II of Volume I of this report.

2.1 Actinic flux and photolysis rate coefficients

Radiation transfer models are used to obtain theoretical estimations of quantities that depend on the radiation flux. Examples of such quantities include, but are not limited to, coefficients determining the speed of chemical reactions (photolysis) induced by solar radiation in the atmosphere, solar radiation flux, radiation effects on living species, and heating of the earth by solar radiation. Two quantities of importance for this purpose are the actinic flux and the irradiance. Both measure the amount of light that crosses a unit surface area per unit of time, and per unit interval of wavelength. The actinic flux is the spherically-integrated light intensity that is appropriate for calculating photolysis reaction rate coefficients (see Madronich, 1987). The irradiance refers to a flux across a horizontal plane; sunlight intensity is commonly measured and reported in terms of irradiance. The actinic flux $F$ and irradiance $I$ as a function of frequency $\nu = hc/\lambda$ are defined by:

$$
F(\nu) \equiv \int d\phi \int d\theta \, u_{\nu}(\theta,\phi) \sin \theta
$$

$$
I(\nu) \equiv \int d\phi \int d\theta \, u_{\nu}(\theta,\phi) \cos \theta \sin \theta
$$

where $u_{\nu}(\theta,\phi)$ is the spectral radiance (sometime also called radiation specific intensity or, simply, intensity) associated with a specific zenith angle $\theta$ and azimuth angle $\phi$.

In the context of air quality modeling, radiation transfer models are of interest for computing photolysis rate coefficients $J$. The rate of chemical species photolysis via a given reaction $X$ is found by multiplying $J_X$ with the concentration of the species undergoing the photolysis. The photolysis rate coefficient $J_X$ is evaluated by integrating over wavelengths using the relationship

$$
J_X = \int_0^\infty \sigma_X(\lambda) \, \varphi_X(\lambda) \, F(\lambda) \, d\lambda
$$

where $\sigma_X(\lambda)$ is the absorption cross section (cm$^2$), $\varphi_X(\lambda)$ is the quantum yield, and $F(\lambda)$ is the actinic flux (photons cm$^{-2}$ s$^{-1}$ nm$^{-1}$). For a specific molecule reacting in the troposphere, the lower
limit of integration is frequently set $\lambda = 290$ nm and the higher limit is the longest wavelength where the photochemical reaction occurs (the reaction threshold). Most photolysis reactions of interest for air quality occur in the wavelength region $\lambda = 290$ to 420 nm (i.e., the near UV).

### 2.2 Radiation transfer general equation

The object of the radiation transfer package described in this manual is to determine how the radiance propagates through the atmosphere. Most radiation transfer models are based on theoretical developments attributed to Chandrasekhar (1960). Chandrasekhar defines the radiance $u_v$ at frequency $v$ by its relationship to “the amount of radiant energy $dE_v$, in a frequency interval $(v, v + dv)$, which is transported across an element of area $d\sigma$ and in directions confined to an element of solid angle $d\omega$, during a time $dt$”:

$$dE_v = u_v \cos \theta \, dv \, d\sigma \, d\omega \, dt$$  \hspace{1cm} (3)$$

When the radiation transfer occurs in an atmosphere composed of parallel horizontal layers, the radiance transfer equation can be expressed as (see Liou, 1973 or Stamnes et al., 1988):

$$\mu \frac{du_v(\tau_v, \mu, \phi)}{d\tau_v} = u_v(\tau_v, \mu, \phi)$$

$$- \frac{\omega_v(\tau_v)}{4\pi} \int_0^{\pi} d\phi' \int_{-1}^{1} d\mu' P_v(\tau_v, \mu, \phi, \mu', \phi') u_v(\tau_v, \mu', \phi')$$

$$- Q_v(\tau_v, \mu, \phi)$$  \hspace{1cm} (4)$$

Eq. (4) describes the radiance transfer in the direction defined by $\mu$ the cosine of the zenith angle and $\phi$ the azimuth angle at the altitude $z$ corresponding to the optical depth $\tau_v$ defined by:

$$\tau_v = \int_z^\infty k_v \rho \, dz$$  \hspace{1cm} (5)$$

where $k_v$ is the extinction coefficient and $\rho$ the density. The first term of the right-hand side of Eq. 4 is a loss term proportional to the radiance. The remainder of the right-hand side is a source term split into a scattering source term and another term, $Q_v$, covering all other sources. The scattering source term expresses the amount of radiance that is scattered from the direction $(\mu', \phi')$ into the direction $(\mu, \phi)$. $P_v(\tau_v, \mu, \phi, \mu', \phi')$ is the phase function, and $\omega_v(\tau_v)$ is the single scattering albedo. While a term for thermal emission in local thermodynamic equilibrium is
sometime included in $Q_ν$, only the contribution from the direct solar beam will be considered here. In this case,

$$Q_ν = \frac{ω_ν(τ_ν)}{4} F_{0ν} P_ν(τ_ν, μ, φ, −μ_0, φ_0) \exp \left( -\frac{τ_ν}{μ_0} \right)$$

(6)

where $πF_0$ is the incident solar flux coming from the direction $(μ_0, φ_0)$. When there is no explicit radiance transfer from one frequency to another (the thermal emission term can be viewed as an implicit transfer of radiance), the frequency index $ν$ can be dropped for simplicity without loss of generality.

Analytical solutions of Eq. (4) can only be found in simplified cases, and numerical methods are often used. The main difficulty comes from the integration over the whole solid angle in the scattering source term. Most methods for finding analytical or numerical solutions to Eq. (4) have common features that were devised by Chandrasekhar. The first step common to almost all methods is to simplify the integration over the azimuth angle in Eq. 4 by expanding the radiance in a Fourier cosine series:

$$u(τ, μ, φ) = \sum_{m=0}^{2N-1} u^m(τ, μ) \cos m(φ_0 − φ),$$

(7)

and the phase function in a sum of Legendre polynomials that, by using the addition theorem for spherical harmonics, is transformed to:

$$P(τ, μ, φ, μ', φ') = \sum_{m=0}^{2N-1} (2 − δ_{0,m}) \cos m(φ' − φ) \sum_{l=m}^{2N-1} (2l + 1) g^m_l(τ) P^m_l(μ) P^m_l(μ')$$

(8)

where $g^m_l$ are sets of constants that are determined using the orthogonal property of Legendre polynomials. A set of $2N$ decoupled equations (one for each Fourier component) is obtained:

$$μ \frac{du^m(τ, μ)}{dτ} = u^m(τ, μ)$$

$$− \frac{ω(τ)}{2} \sum_{l=m}^{2N-1} (2l + 1) g^m_l(τ) P^m_l(μ) \int_{−1}^{1} dμ' P^m_l(μ') u^m(τ, μ')$$

$$− \frac{ω(τ)}{4} F_0 \exp \left( -\frac{τ}{μ_0} \right) (2 − δ_{0,m}) \sum_{l=m}^{2N-1} (−1)^{l+m} (2l + 1) g^m_l(τ) P^m_l(μ) P^m_l(μ_0)$$

(9)

2.3 Two-stream methods

In two-stream methods (see Meador and Weaver, 1980), it is assumed that only two streams of radiation (i.e., two directions) exist. Both streams are vertical, one going downward, and the
other upward. In such a case, the solution is symmetric with respect of the azimuth angle, and only the equation for \( m = 0 \) in Eq. (9) should be retained. Defining \( p(τ, µ, µ') \) as:

\[
p(τ, µ, µ') = \omega(τ) \sum_{l=0}^{2N-1} (2l + 1) g_l^0(τ) P_l^0(µ) P_l^0(µ'),
\]

the following transfer equation is obtained (the superscript index \( m=0 \) is dropped for simplicity):

\[
\frac{d u(τ, µ)}{d τ} = u(τ, µ) - \frac{1}{2} \int_{-1}^{1} dµ' p(τ, µ, µ') u(τ, µ') - \frac{1}{4} F_0 \exp\left(-\frac{τ}{µ_0}\right) p(τ, µ, -µ_0).
\]

The streams are defined as the following hemispheric integrals:

\[
u_±(τ) = \int_{0}^{1} u_±(τ, ±µ) µ dµ,
\]

and the quantity \( β_0 \) is defined as:

\[
β_0(τ) = \frac{1}{2ω(τ)} \int_{0}^{1} p(τ, µ, -µ_0) dµ.
\]

Because the phase function is normalized as \( \int_{-1}^{1} p(τ, µ, µ') dµ' = 2ω(τ) \), then

\[
β_0(τ) = 1 - \frac{1}{2ω(τ)} \int_{0}^{1} p(τ, µ, -µ_0) dµ.
\]

Thus, after integrating Eq. (11) from 0 to 1 along \( µ \) in one case and \(-µ\) in the other, the following pair of equation is obtained:

\[
\frac{d u^+(τ)}{d τ} = \int_{0}^{1} u(τ, µ) dµ - \frac{1}{2} \int_{0}^{1} dµ' \int_{-1}^{1} dµ'' p(τ, µ, µ') u(τ, µ') - \frac{1}{2} F_0 ω(τ) β_0(τ) \exp\left(-\frac{τ}{µ_0}\right)
\]

\[
\frac{d u^-(τ)}{d τ} = -\int_{0}^{1} u(τ, -µ) dµ + \frac{1}{2} \int_{0}^{1} dµ' \int_{-1}^{1} dµ'' p(τ, -µ, µ') u(τ, µ') + \frac{1}{2} F_0 ω(τ) (1 - β_0(τ)) \exp\left(-\frac{τ}{µ_0}\right)
\]

The two-stream methods approximate \( u(τ, µ) \) so that it is related to \( u^+(τ) \) to obtain the following general form for the transfer equation:
\[
\frac{du^+}{d\tau} = \gamma_1 u^+(\tau) - \gamma_2 u^-(\tau) - F_0 \omega(\tau) \gamma_3 \exp\left(-\frac{\tau}{\mu_0}\right)
\]

\[
\frac{du^-}{d\tau} = \gamma_2 u^+(\tau) - \gamma_1 u^-(\tau) + F_0 \omega(\tau) \gamma_4 \exp\left(-\frac{\tau}{\mu_0}\right)
\]  

(16)

The parameters \(\gamma_1\), \(\gamma_2\), \(\gamma_3\), and \(\gamma_4\) depend on the assumed functional form \(p(\tau,\mu,\mu')\). For the most commonly chosen functional forms, \(\gamma_1\), \(\gamma_2\), \(\gamma_3\), and \(\gamma_4\) are simple functions of the single scattering albedo \(\omega(\tau)\), the asymmetry parameter \(g(\tau)\) and \(\mu_0\). Consequently, the \(\gamma\) parameters depend on \(\tau\). In order to solve the coupled Eqs. (16), the atmosphere is divided in thin horizontal layers where the parameters are assumed constant\(^1\). Within a layer, Toon et al. (1989) have shown that the general solution to the coupled Eqs. (16) is:

\[
u^i_+ (\tau) = k_1 \exp(\lambda_i \tau) - \Gamma_i k_2 \exp(-\lambda_i \tau) + C^+_i (\tau)
\]

\[
u^i_-(\tau) = \Gamma_i k_1 \exp(\lambda_i \tau) - k_2 \exp(-\lambda_i \tau) + C^-_i (\tau)
\]  

(17)

where the index \(i\) stands for the layer number, \(k_1\) and \(k_2\) depend on the boundary conditions, and by assuming a layer of optical depth \(\tau_i\) and cumulative optical depth \(\tau_c\) for the layers above, \n
\[
\lambda_i = \sqrt{\gamma_1^2 - \gamma_2^2}
\]

\[
\Gamma_i = \frac{\gamma_2}{\gamma_1 + \lambda_i}
\]

\[
C^+_i (\tau) = \frac{\omega_i \pi F_0 \left[ (\gamma_1^2 - 1/\mu_0) \gamma_3 + \gamma_4 \gamma_2^2 \right] \exp\left(-\tau_i + \tau\right)}{\lambda_i^2 - 1/\mu_0}
\]

\[
C^-_i (\tau) = \frac{\omega_i \pi F_0 \left[ (\gamma_1^2 + 1/\mu_0) \gamma_3 + \gamma_4 \gamma_2 \right] \exp\left(-\tau_i + \tau\right)}{\lambda_i^2 - 1/\mu_0}
\]

(18)

First, Toon et al. scaled the terms so that none of them includes a positive \(\tau\)-dependent exponent (terms with such positive exponents can lead to numerical instabilities):

\(^1\) When deriving the solution, the layers are assumed to be planar. Corrections are made afterward to take into account the atmosphere sphericity. Such corrections are only important for large solar zenith angle.
\[ u_i^-(\tau) = Y_{i1} \left( \exp[-\lambda_i (\tau - \tau)] + \Gamma_i \exp[-\lambda_i \tau] \right) + Y_{i2} \left( \exp[-\lambda_i (\tau - \tau)] - \Gamma_i \exp[-\lambda_i \tau] \right) - C_i^+ (\tau) \]
\[ u_i^+(\tau) = Y_{i1} \left( \Gamma_i \exp[-\lambda_i (\tau - \tau)] + \exp[-\lambda_i \tau] \right) + Y_{i2} \left( \Gamma_i \exp[-\lambda_i (\tau - \tau)] - \exp[-\lambda_i \tau] \right) + C_i^- (\tau) \]

where \( \tau \) is the total optical depth of the layer, and \( \tau_i \) is a parameter that depends on the chosen two-stream approximation. Toon et al. (1989) gave the value of these parameters for different type of two-stream methods. The method used in TUVAQM is the delta-scaled Eddington approximation. Code for other approximations is also included but inactivated (commented out). To use one of these other approximations, the parameter \( \mu_i \) is a parameter that depends on the chosen two-stream approximation.
approximations, the user needs to make code changes in SUBROUTINE PS2STR in file rad_tr.f (see section 8).

2.4 Other methods

One version of the original Tropospheric Ultraviolet Visible (TUV) radiation transfer program uses the discrete ordinate method. This method and other related methods are more complex than two-stream approximations, and allow a more precise description of the angular dependence of the radiance. They rely on applying a Gaussian quadrature rule to integrate over the zenith angle following Eq. (9). (For details, see Chandrasekhar, 1960, Liou, 1973 or Stamnes et al., 1988.)

The integral over $\mu$ is replaced by a discrete sum over quadrature zenith angle using Gauss quadrature. For each of the $2N$ equations for the Fourier component a set of $2n$ coupled equations should be solved (one for each of the quadrature zenith angles):

$$
\mu_i \frac{du^m(\tau, \mu_i)}{d\tau} = u^m(\tau, \mu_i)
- \frac{\omega(\tau)}{2} \sum_{l=m}^{2N-1} (2l + 1) g^m_i(\tau) P^m_i(\mu_i) \sum_{j=-n}^{n} a_j P^m_j(\mu_j) u^m(\tau, \mu_j)
- \frac{\omega(\tau)}{4\pi} I_0 \exp \left( -\frac{\tau}{\mu_0} \right) (2 - \delta_{0,m}) \times
\sum_{l=m}^{2N-1} (-1)^{i+m} (2l + 1) g^m_i(\tau) P^m_i(\mu_i) P^m_i(\mu_0) \quad i = \pm 1, \ldots, \pm n
$$

(24)

The discrete-ordinate method and the other related methods allow an arbitrarily precise description of the radiance angular dependence, according to the number of terms considered. At the lowest order ($m = 0$, and $n = 1$), they reduce to a two-stream method. When good precision in the angular description is desired, the dimensionality of the equation system is large. There are $M \times 2n \times 2N$ equations, where $M$ is the number of atmospheric horizontal layers, $2n$ is the number of Gaussian quadrature zenith angles, and $2N$ is the number of Fourier azimuth components.

3 Using the library within an AQM

TUVAQM can be used in the framework on an air quality model (AQM) for determining rate coefficients for the photolysis reactions. For using TUVAQM, the main AQM program needs only to call a single generic routine (subroutine tuvaqm). This routine can be called in an initialization mode (hopefully once in the AQM program execution) or in the normal execution.

---

2 The number of Fourier components is usually restricted by assumptions over the functional form of the phase function.
mode (every time a photolysis reaction rate coefficient evaluation is needed in the AQM). Many factors influence radiation transfer in the atmosphere. These factors are parameters that are inputs to the program and their values can be set via input files. The following parameters must be defined in **TUVAQM**:

1) General parameters such as date, time, time zone, longitude, latitude, and zenith angle.
2) Elevation grid, i.e., the vertical column structure the program uses for the computations.
3) Wavelength grid, i.e., the wavelength bins the program uses for the computations.
4) Air pressure.
5) Absorption and scattering by aerosols.
6) Absorption and scattering by clouds.
7) Ground albedo.
8) Air temperature.
9) Absorption by ozone.
10) Absorption by SO$_2$.
11) Absorption by NO$_2$.
12) Extraterrestrial solar flux.

Reasonable default values are provided for these parameters, and the user can change the values using a general input file read at initialization time (see section 5). Values set this way will be used for the whole AQM simulation. However, multiple situations can occur depending on the modeling domain cell (location) and modeling time. Thus, some parameters may apply throughout the whole domain and modeling period, while some others should be allowed to vary from cell to cell and time to time. Unfortunately, data that can be used to specify these parameters are usually scarce, and the spatial and temporal resolution is often poor. The strategy chosen in the current implementation is to select a reduced number of parameters that can vary within the AQM simulation domain and episode. These parameters are:

1) zenith angle,
2) elevation grid,
3) ground albedo,
4) total ozone column,
5) total aerosol optical depth,
6) aerosol single scattering albedo, and
7) aerosol asymmetry factor.
The user *must* specify the temporal and spatial dependence for these values. First, the user must divide the modeling domain into regions and the modeling episode into time intervals such that the atmospheric properties can be considered as constant within a given region and time interval. Then the user needs to specify the parameter values for each region and time interval using the input file described in section 6.

For each region and time interval, a set of complete radiation transfer calculations will be performed for conditions representative of the region and time interval. For all AQM spatial cells within a region, and time steps within an interval, the photolysis reaction rate coefficients will be computed by interpolation from the full radiation transfer calculations.

4 Using the library with a standalone program

TUVAQM consists of two classes of routines: generic routines that perform groups of tasks and basic routines that perform a single task. Typically, the generic routines use the basic routines to perform groups of tasks such as initialization, or a full radiation transfer calculation. The generic routines can be used to create a standalone radiation transfer simulation program. Examples of such programs are provided, and the instructions for installing them are given in section 7. These examples include one program for performing a single simulation and another for repeated simulations with a reduced set of parameters (up to five) varied in a systematic way for sensitivity studies.

The example programs read an input file for setting the simulation conditions (the parameters listed in section 3), compute one or more simulations and, depending on the version (see section 7), save the results of the simulations. The following section describes the syntax of the input file.

The library routines can also be used for creating a new radiation transfer simulation program. The section 8 gives an overview of the test program as well as the generic and basic libraries.

5 General card and command description

The general card and command input file allows changing the default values of the parameters describing the radiation transfer situation being simulated. Technically, this means setting values for code variables or variable arrays. In simple cases, a *card* key word is followed by a list of values. In more complex cases, when sets of values need to be changed in a coherent way, a *command* key word is followed by other key words and values.
5.1 Card description and syntax

A card is a segment of the input file (possibly extending over several lines) that starts with a card key word in the first four columns of the first line and is followed by a list of values ending when the next key word is encountered. Each card is used to set the value(s) for one parameter assigned in the computer code to one variable or variable array. Consequently each card key word is linked to one code variable of a given type and length. The types can be logical, integer, real, double precision, and character strings. Character strings are restricted to strings with a multiple of four characters. The length is equal to the number of elements of the corresponding variable array.

Example

TEMD 288.150 281.651 275.154

The card key word is TEMD (linked to the internal variable array TMPDEF) that is used to define the atmospheric temperature profile. In the example the three lowest points of the temperature elevation grid are given the values 288.150, 281.651 and 275.154 K.

The list of values consists of values separated by one or more blank characters or tabulation characters. In order to facilitate setting values for an array, locators and repetitors can be used. A locator is an integer index followed by an equal sign (=) immediately preceding a value. The locator indicates that the following value is for the given index in the variable array corresponding to the card key word. A repetitor is an integer number n followed by a star sign (*) immediately preceding a value. The repetitor indicates that the given value is for n indices of the variable array, starting from the current index.

Example

TEMD 3*290.0 6*280 25=273.15 4*2.73E+02 15=278 275.

Indices 1, 2 and 3 of variable TMPDEF have the value 290 K, indices 4 to 9 have the value 280 K, index 25 has the value 273.15 K, indices 26 to 29 have the value 273 K, index 15 has the value 278 K and index 16 has the value 275 K. As shown in the example, indices can be skipped, locators can be given in any order, and indices increase monotonically starting from the last given locator or 1 if no previous locator is given.

In order to increase the readability of the input file, comment lines starting with an exclamation mark can be included at any time,
Example

! Number of elevation grid levels
! NBZG 46
! Defines elevation grid levels (kilometers above sea level)
!
! height  top of cell
! elevation
! ground elevation:  260 m
! cell 1  50 m  310 m
! cell 2  100 m  410 m
! cell 3  300 m  710 m
! cell 4  1290 m  2000 m
! cell 5  2000 m  4000 m
! cell 6  3000 m  7000 m
! cell 7  5000 m  12000 m
! cell 8-46  1000 m to 50000 m
!
! Cell 1-7
ZGRD 0.26 0.31 0.41 0.71 2.00 4.00 7.00
! Cell 8-46
   12.00 13.00 14.00 15.00 16.00 17.00 18.00 19.00 20.00
   21.00 22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00
   31.00 32.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00 40.00
   41.00 42.00 43.00 44.00 45.00 46.00 47.00 48.00 49.00 50.00

NBZG and ZGRD are linked to variables describing the program general elevation grid. The number of elevations in the grid is set to 46, and the corresponding elevations are given. Comments are included to explain the grid structure.

The format for the values is the following:

- Logical values are "._TRUE._" or "._FALSE._".
- Numbers can be given in integer, floating point or exponential notation.
- Character strings should be single quoted.

Cards can be omitted and values can be skipped within a card. In such a case, the default value is used. Cards can be given in any order, and can appear more than once. In case the same index of the same card is defined more than once, the last defined value is used. A description of all the recognized card key words with the corresponding variable, type and dimension, as well as the default values and description of the use of the variable in TUVAQM is given below.
5.1.1 Card list

5.1.1.1 General parameters:

The following variables are used to define general conditions of the simulation such as time and location for the simulations. It is possible to request TUVAQM to determine the solar zenith angle or to define the zenith angle as an input parameter. When TUVAQM determines the zenith angle, the location and time are important. When the zenith angle is an input parameter, time and location are ignored.

- Card key word: LIST
  
  Code variable: LISCRD Type: LOGICAL Dimension: 1
  
  Description:
  Logical switch for printing user’s card value. When this card is set to .TRUE., the card values are printed after the input file is read.
  
  Default: .FALSE.

- Card key word: DATE
  
  Code variable: IDATE Type: INTEGER Dimension: 1
  
  Description:
  Decimal date (YYMMDD). This variable is used when one wants TUVAQM to compute the zenith angle. If the zenith angle is provided, this variable has no influence on the library routines.
  
  Default: 970701

- Card key word: LATI
  
  Code variable: ALAT Type: REAL Dimension: 1
  
  Description:
  Latitude (degree). This variable is used when one wants TUVAQM to compute the zenith angle. If the zenith angle is provided, this variable has no influence on the library routines. Latitude is between 0 and 90° for the Northern Hemisphere, and 0 and -90° for the Southern Hemisphere.
  
  Default: 34.0
- Card key word: **LONG**
  Code variable: **ALONG** Type: **REAL** Dimension: 1
  Description:
  Longitude (degree). This variable is used when one wants **TUVAQM** to compute the zenith angle. If the zenith angle is provided, this variable has no influence on the library routines. The longitude is between 0 and 180° for longitudes east of Greenwich Meridian, and between 0 and -180° for longitudes west of Greenwich Meridian.
  Default: -117.3

- Card key word: **UTSH**
  Code variable: **UT0** Type: **REAL** Dimension: 1
  Description:
  Universal time shift (hours), i.e., the time difference between GMT and the local time. This variable is used when one wants **TUVAQM** to compute the zenith angle. If the zenith angle is provided, this variable has no influence on the library routines. This variable is used for conversion between the local time zone and the universal time. Regions west of Greenwich Meridian need a positive value, and region east of Greenwich Meridian a negative positive one. **UTSH** is 8 for Pacific Standard Time, and 7 for Pacific Daylight Time.
  Default: 8.5

- Card key word: **LTIM**
  Code variable: **LUTIME** Type: **LOGICAL** Dimension: 1
  Description:
  Logical switch for zenith angle computation. When this card is set to .TRUE., the zenith angle is computed at the desired time steps, given the location and date.
  Default: .FALSE.

- Card key word: **NSTO**
  Code variable: **NBSTOP** Type: **INTEGER** Dimension: 1
  Description:
  Number of times or zenith angles at which radiation transfer computations are requested.
  Default: 13
• Card key word: SPNT  
  Code variable: S_PNTS  Type: REAL  
  Dimension: MAX_ST = 100  
  Description: 
  Times or zenith angles at which radiation transfer computations are requested. 
  Default: 0.0, 5.0, 10.0, 20.0, 30.0, 40.0, 50.0, 60.0, 70.0, 80.0, 85.0, 88.0, 90.0, 87*0.0

5.1.1.2 Wavelength and elevation general grid choice:

The following variables are used in SUBROUTINE TUV_RD. They define the segmentation in wavelength and elevation that TUIVAQM uses to compute solar irradiance. Many inputs are given on their own wavelength or elevation grid. These grids do not coincide with the general wavelength or elevation grid. Library routines interpolate the input grids on the general grid.

• Card key word: CHWL  
  Code variable: MOPT_GW  Type: INTEGER  
  Dimension: 1  
  Description: 
  Flag designing which wavelength grid is used. The options are: 
  1 = equal spacing 
  2 = Isaksen's grid 
  3 = Combined Kockarts/Isaksen’s grid 
  4 = User defined 
  Default: 2  (Note: option 4 requires recoding)

• Card key word: NBZG  
  Code variable: NZ  Type: INTEGER  
  Dimension: 1  
  Description: 
  Number of elevation cell boundaries (i.e., the number of elevation cells plus 1) in the general elevation grid. In case NZ = 0, a regular elevation grid with 51 levels starting at 0 km and ending at 50 km is chosen. 
  Default: 0  (i.e., 51 levels.)
• Card key word: ZGRD
  Code variable: Z Type: REAL Dimension: KZ = 101

  Description:
  Elevation cell boundaries in the general elevation grid. In case NZ = 0 (see NBZG),
a regular elevation grid with 51 levels starting at 0 km and ending at 50 km is
chosen. If the elevation grid is defined by the user, it must be in strictly ascending
order.

  Default: KZ * 0 i.e. 51 levels {0,1,2,3,…,50} (see NBZG)

5.1.1.3 Air pressure (air density) definition:

  The following variables are used in SUBROUTINE SETAIR. The air density is given for
layers of the atmosphere where the density is assumed to be constant.

• Card key word: APTI
  Code variable: APRTIT Type: CHARACTER*32 Dimension: 1

  Description:
  Title for the air pressure elevation profile definition. This variable does not
influence the radiation transfer simulation and is only used when input values are
printed out.

  Default: ’air density: USSA, 1976’

• Card key word: APRN
  Code variable: APRNEW Type: REAL Dimension: 1

  Description:
  Sea level air pressure. An air density elevation profile can be defined (see below).
In general, the sea level air pressure is defined by summing the air density
elevation profile. If a sea level pressure is given, the air density profile is adjusted
(scaled) such that the sea level air pressure is equal to APRNEW. When APRNEW is
less than zero, the sea level pressure is determined by summing the air density
profile.

  Default: -999.0 i.e., sea level pressure defined by air density elevation profile.
• Card key word: APNZ
  Code variable: NBAPRZ  Type: INTEGER  Dimension: 1
  Description:
  Number of air pressure elevation profile grid cell limits.
  Default: 121

• Card key word: APZG
  Code variable: ZG_APR  Type: REAL  Dimension: KZGAPR = 151
  Description:
  Air density elevation profile grid, i.e., levels at which the air density is given. The air density elevation grid must be in a strictly ascending order. The elevation grid is given in kilometer.
  Default: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
  11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
  21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
  31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
  41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
  51, 52, 53, 54, 55, 56, 57, 58, 59, 60,
  61, 62, 63, 64, 65, 66, 67, 68, 69, 70,
  71, 72, 73, 74, 75, 76, 77, 78, 79, 80,
  81, 82, 83, 84, 85, 86, 87, 88, 89, 90,
  91, 92, 93, 94, 95, 96, 97, 98, 99, 100,
  101, 102, 103, 104, 105, 106, 107, 108, 109, 110,
  111, 112, 113, 114, 115, 116, 117, 118, 119, 120,
  30*1

• Card key word: APRD
  Code variable: APRDEF  Type: REAL  Dimension: KZGAPR = 151
  Description:
  Air density elevation profile, i.e., air density at altitude levels of elevation grid. Default values are annual means from US Standard Atmosphere, 1976, for 45N (See NOAA, 1976). Values are number density (cm^3).
  Default: 2.55E+19, 2.31E+19, 2.09E+19, 1.89E+19, 1.70E+19,
  1.53E+19, 1.37E+19, 1.23E+19, 1.09E+19, 9.71E+18,
  8.60E+18, 7.59E+18, 6.49E+18, 5.54E+18, 4.74E+18,
5.1.1.4 Scaling and interpolation

Several of the atmospheric optical parameters described below are interpolated and can be scaled. The interpolation is done because the parameters are each given on an elevation profile grid that is specific to the parameter considered. For performing the radiative transfer calculations, TUVAQM interpolates the results given on the specific grids to its general grid. When a specific grid is larger than the general grid (e.g., a parameter is given from the sea level to the
top of the atmosphere, and the general grid starts from a ground elevation above sea level), part of
the specific grid is chopped off to fit into the general grid. The scaling is done to change an
elevation profile so that its sum corresponds to a given value. For example, a default aerosol
optical depth elevation profile is given. It is possible to give a total aerosol optical depth, and
request that the aerosol optical depth of each elevation grid cell will be scaled so that the total
aerosol optical depth (the sum over the elevation cells) will correspond to the given value. In
such a case, it is important to know if the scaling is done before or after the interpolation. In case
a part of the specific elevation grid is chopped off, scaling done before interpolation will consider
the full sum, while scaling done after interpolation will consider the partial sum fitting into the
general grid. Both options can be valid. If a quantity is usually given for the whole atmosphere
(e.g., sea-level air pressure or ozone total column), it makes sense to scale before interpolation. If
a quantity is usually given at a given site considering only the portion of the atmosphere above
the given site, it makes sense to scale after interpolation. For each of the following quantity that
can be scaled, a note indicates whether the scaling is done before or after the interpolation.
Changing this order can be accomplished only be changing the computer code (see section 8).

5.1.1.5 Aerosol definition:

The following variables are used in SUBROUTINE SETAER. The aerosol profile is given
for atmospheric elevation layers in which the aerosol density and other properties are assumed
constant. The aerosol optical depth per kilometer at 340 nm is given for each layer. For
computing the aerosol optical depth at other wavelengths, TUVAQM assumes the optical depth to
be inversely proportional to the first power of the wavelength:

$$\tau(\lambda) = \frac{\lambda_{340}}{\lambda} \tau(\lambda_{340})$$

where $\tau$ is the optical depth and $\lambda$ the wavelength.

- Card key word: AETI
  Code variable: AERTIT  Type: CHARACTER*32  Dimension: 1
  Description:
  Title for the aerosol elevation profile definition. This variable does not influence
  the radiation transfer simulation and is only used when input values are printed out.
  Default: 'Aerosols: Elterman (1968) '
• Card key word: AERF
  Code variable: AEROSL  Type: LOGICAL  Dimension: 1
  Description:
  Logical switch for using aerosol. If the switch is set to .FALSE., the aerosol total
  optical depth is automatically set to zero.
  Default: .FALSE.

• Card key word: AODN
  Code variable: AODNEW  Type: REAL  Dimension: 1
  Description:
  Total vertical aerosol optical depth. An aerosol optical depth elevation profile can
  be defined by specifying the aerosol optical depth of each aerosol elevation profile
cell (see below). In general, the total vertical aerosol optical depth is defined by
summing the aerosol optical depths over the elevation profile. If a total vertical
aerosol optical depth is given, the aerosol elevation cell optical depths are adjusted
(scaled) such that the total vertical aerosol optical depth is equal to AODNEW.
When AODNEW is less than zero, the total vertical aerosol optical depth is
determined by summing the aerosol optical depths over the elevation profile.
AODNEW is the total optical depth for light at wavelength = 340 nm.
  Note: Because the aerosol elevation grid is interpolated on the general elevation grid, it
matters whether the scaling is performed before or after the interpolation (see
section 5.1.1.4). The scaling is done on the full aerosol profile before interpolation,
and not on the aerosol profile on the general grid.
  Default: -999.0  i.e., total aerosol optical depth defined by its elevation profile.

• Card key word: AENZ
  Code variable: NBAERZ  Type: INTEGER  Dimension: 1
  Description:
  Number of elevation cell boundaries (i.e., the number of elevation cells plus 1) for
the aerosol elevation profile grid.
  Default: 51
Card key word: AEZG  
Code variable: ZG_AER  Type: REAL  Dimension: KZGAER = 101

Description:
Aerosol elevation profile grid, i.e., levels at which the aerosol optical depth and other properties are given. The aerosol elevation grid must be in a strictly ascending order. The elevation grid is given in kilometers.

Default: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 50*-1

Card key word: AODD  
Code variable: AODDEF  Type: REAL  Dimension: KZGAER = 101

Description:
Aerosol optical depth elevation profile, i.e., aerosol optical depth per kilometer at altitude levels of the aerosol elevation grid. Values are from Elterman (1968). These are vertical optical depths per km, in 1 km intervals from 0 km to 50 km, for light at wavelength = 340 nm. The total aerosol optical depth at 340 nm, assuming a linear behavior between elevation points, is 0.38.

Default: 2.40E-01, 1.06E-01, 4.56E-02, 1.91E-02, 1.01E-02, 7.63E-03, 5.38E-03, 5.00E-03, 5.15E-03, 4.94E-03, 4.82E-03, 4.51E-03, 4.74E-03, 4.37E-03, 4.28E-03, 4.03E-03, 3.83E-03, 3.78E-03, 3.88E-03, 3.08E-03, 2.26E-03, 1.64E-03, 1.23E-03, 9.45E-04, 7.49E-04, 6.30E-04, 5.50E-04, 4.21E-04, 3.22E-04, 2.48E-04, 1.90E-04, 1.45E-04, 1.11E-04, 8.51E-05, 6.52E-05, 5.00E-05, 3.83E-05, 2.93E-05, 2.25E-05, 1.72E-05, 1.32E-05, 1.01E-05, 7.72E-06, 5.91E-06, 4.53E-06, 3.46E-06, 2.66E-06, 2.04E-06, 1.56E-06, 1.19E-06, 9.14E-07, 50*0
• Card key word: AOMD
  Code variable: AOMDEF  Type: REAL    Dimension: KZGAER = 101
  Description:
  Aerosol single scattering albedo elevation profile, i.e., aerosol single scattering
  albedo at altitude levels of the aerosol elevation grid.
  Default: KZGAER * 0.99

• Card key word: AGDE
  Code variable: AG_DEF  Type: REAL    Dimension: KZGAER = 101
  Description:
  Aerosol asymmetry factor elevation profile, i.e., aerosol asymmetry factor at
  altitude levels of the aerosol elevation grid.
  Default: KZGAER * 0.61

5.1.1.6 Cloud definition:

The following variables are used in SUBROUTINE SETCLD. Since TUVAQM considers only
one spatial dimension (elevation), partial cloud coverage and the effects of cloud edges cannot be
represented in this model. It is possible to include some cloud layers that are treated in a way
similar to the aerosols. Four cloud layers are defined; however, the default is set to simulate a
clear sky situation (all cloud layers have zero optical depths).

• Card key word: CLTI
  Code variable: CLDTIT  Type: CHARACTER*32  Dimension: 1
  Description:
  Title for the cloud elevation profile definition. This variable does not influence the
  radiation transfer simulation and is only used when input values are printed out.
  Default: ‘Default zero cloud optical depth’

• Card key word: CLNZ
  Code variable: NBCLDZ  Type: INTEGER  Dimension: 1
  Description:
  Number of elevation cell boundaries (i.e., the number of elevation cells plus 1) for
  the cloud elevation profile grid.
  Default: 4
• Card key word: CLZG
  Code variable: ZG_CLD  Type: REAL  Dimension: KZGCLD = 21
  Description:
  Cloud elevation profile grid, i.e., levels at which the cloud optical depth and other
  properties are given. The cloud elevation grid must be in a strictly ascending order.
  The elevation grid is given in kilometers.
  Default: 5, 7, 9, 11, 17*0

• Card key word: CODD
  Code variable: CODDEF  Type: REAL  Dimension: KZGCLD = 21
  Description:
  Cloud optical depth elevation profile, i.e., cloud optical depth per kilometer at
  altitude levels of the cloud elevation grid. All optical depths are set to zero for
  simulating a clear-sky situation.
  Default: KZGCLD * 0

• Card key word: COMD
  Code variable: COMDEF  Type: REAL  Dimension: KZGCLD = 21
  Description:
  Cloud single scattering albedo elevation profile, i.e., cloud single scattering albedo
  in cells of the cloud elevation grid.
  Default: 0.9999, 0.5, 0.9999, 18*0

• Card key word: CGDE
  Code variable: CG_DEF  Type: REAL  Dimension: KZGCLD = 21
  Description:
  Cloud asymmetry factor elevation profile, i.e., cloud asymmetry factor in cells of
  the cloud elevation grid.
  Default: 0.85, 0.5, 0.85, 18*0

5.1.1.7 Ground albedo definition:

The following variables are used in SUBROUTINE SETALB. The ground albedo can be
defined on wavelength bins. The default is to define a ground albedo valid at all wavelengths.
• Card key word: ALBT
  Code variable: ALBTIT  Type: CHARACTER*32  Dimension: 1
  Description:
  Title for the ground albedo definition. This variable does not influence the radiation transfer simulation and is only used when input values are printed out.
  Default: ’Wavelength-independent albedo ’

• Card key word: ALNW
  Code variable: NBALBW  Type: INTEGER  Dimension: 1
  Description:
  Number of ground albedo wavelength profile grid bins.
  Default: 1

• Card key word: ALWG
  Code variable: WG_ALB  Type: REAL  Dimension: KWGALB = 100
  Description:
  Ground albedo wavelength profile grid, i.e., wavelengths at which the ground albedo is given. The ground albedo wavelength grid must be in a strictly ascending order. The wavelength grid is given in nanometers.
  Default: 0.0, 1000.0, 98*–1

• Card key word: ALBD
  Code variable: ALBDEF  Type: REAL  Dimension: KWGALB = 100
  Description:
  Ground albedo wavelength profile, i.e., ground albedo at boundaries of wavelength grid bins. The default is a wavelength-independent ground albedo defined on one bin that covers the whole wavelength range.
  Default: 0.10, 0.10, 98*0

5.1.1.8 Atmospheric temperature definition:

The following variables are used in SUBROUTINE SETTMP. The air density is given for layers of the atmosphere where the density is assumed to be constant.
• Card key word: TMTI
  Code variable: TMPTIT Type: CHARACTER*32 Dimension: 1
  Description:
  Title for the atmospheric temperature elevation profile definition. This variable does not influence the radiation transfer simulation and is only used when input values are printed out.
  Default: 'air temperature: USSA, 1976'

• Card key word: TMNZ
  Code variable: NBTMPZ Type: INTEGER Dimension: 1
  Description:
  Number of elevation cell boundaries (i.e., the number of elevation cells plus 1) for the atmospheric temperature elevation profile.
  Default: 121

• Card key word: TMZG
  Code variable: ZG_TMP Type: REAL Dimension: KZGTMP = 201
  Description:
  Atmospheric temperature elevation profile grid, i.e., levels at which the atmospheric temperature is given. The atmospheric temperature elevation grid must be in a strictly ascending order. The elevation grid is given in kilometers.
  Default: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 80*-1
Card key word: TEMD
Code variable: TMPDEF Type: REAL Dimension: KZGTMp = 201

Description:
Atmospheric temperature elevation profile, i.e., atmospheric temperature at altitude levels of elevation grid. Values are annual means from US Standard Atmosphere, 1976, for 45N (See NOAA, 1976). Temperatures are in Kelvin (K).

Default: 288.150, 281.651, 275.154, 268.659, 262.166, 255.676, 249.187, 242.700, 236.215, 229.733, 223.252, 216.774, 216.650, 216.650, 216.650, 216.650, 216.650, 216.650, 217.581, 218.574, 219.567, 220.560, 221.552, 222.544, 223.536, 224.527, 225.518, 226.509, 227.500, 228.490, 230.973, 233.743, 236.513, 239.282, 242.050, 244.818, 247.584, 250.350, 253.114, 255.878, 258.641, 261.403, 264.164, 266.925, 269.684, 270.650, 270.650, 270.650, 270.650, 269.031, 266.277, 263.524, 260.771, 258.019, 255.268, 252.518, 249.769, 247.021, 244.274, 241.524, 238.781, 236.036, 233.292, 230.549, 227.807, 225.065, 222.325, 219.585, 216.846, 214.263, 212.308, 210.353, 208.399, 206.446, 204.493, 202.541, 200.590, 198.639, 196.688, 194.739, 192.790, 190.841, 188.893, 186.87, 184.87, 182.87, 180.87, 178.87, 176.87, 174.87, 172.87, 170.87, 168.87, 166.87, 164.87, 162.87, 160.87, 158.87, 156.87, 154.87, 152.87, 150.87, 148.87, 146.87, 144.87, 142.87, 140.87, 138.87, 136.87, 134.87, 132.87, 130.87, 128.87, 126.87, 124.87, 122.87, 120.87, 118.87, 116.87, 114.87, 112.87, 110.87, 108.87, 106.87, 104.87, 102.87, 100.87, 98.87, 96.87, 94.87, 92.87, 90.87, 88.87, 86.87, 84.87, 82.87, 80.87, 78.87, 76.87, 74.87, 72.87, 70.87, 68.87, 66.87, 64.87, 62.87, 60.87, 58.87, 56.87, 54.87, 52.87, 50.87, 48.87, 46.87, 44.87, 42.87, 40.87, 38.87, 36.87, 34.87, 32.87, 30.87, 28.87, 26.87, 24.87, 22.87, 20.87, 18.87, 16.87, 14.87, 12.87, 10.87, 8.87, 6.87, 4.87, 2.87, 0.87

5.1.1.9 Ozone elevation profile definition:

The following variables are used in SUBROUTINE SETOZO. The ozone density is given for layers of the atmosphere where the density is assumed to be constant.
• Card key word: O3_T
  Code variable: OZOTIT  Type: CHARACTER*32  Dimension: 1
  Description:
  Title for the ozone elevation profile definition. This variable does not influence the
  radiation transfer simulation and is only used when input values are printed out.
  Default: ‘ozone profile: USSA, 1976’

• Card key word: O3_N
  Code variable: OZONEW  Type: REAL  Dimension: 1
  Description:
  Overhead ozone total column in Dobson Units (DU). One DU is the thickness,
  measured in units of hundredths of a millimeter, that the ozone column would
  occupy at standard temperature and pressure. \( 1 \text{ DU} \approx 2.69 \cdot 10^{16} \text{ molecules cm}^2 \).
  An ozone density elevation profile can be defined (see below). For every cell of
  the ozone elevation profile, the ozone density is converted to a “partial ozone
  column”, and the default ozone total column is defined by summing all the ozone
  partial columns. If a new ozone column is given, the ozone density profile is
  adjusted (scaled) such that the ozone total column is equal to \( OZONEW \). When
  \( OZONEW \) is less than zero, the ozone column is determined by summing the ozone
  partial columns.
  Note: Because the ozone elevation grid is interpolated on the general elevation grid, it
  matters whether the scaling is performed before or after the interpolation (see
  section 5.1.1.4). The scaling is done on the profile on the general grid after
  interpolation, and not on the original ozone profile before interpolation.
  Default: \(-999.0\)  i.e., ozone column defined by ozone density elevation profile.

• Card key word: O3NZ
  Code variable: NBOZOZ  Type: INTEGER  Dimension: 1
  Description:
  Number of elevation cell boundaries (i.e., the number of elevation cells plus 1) for
  the ozone elevation profile.
  Default: 39
• Card key word: O3ZG
  Code variable: ZG_OZO  Type: REAL  Dimension: KZGOZO = 51
  Description:
  Ozone density elevation profile grid, i.e., levels at which the ozone density is given. The ozone density elevation grid must be in a strictly ascending order. The elevation grid is given in kilometers.
  Default: 0, 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 12*1

• Card key word: O3_D
  Code variable: OZODEF  Type: REAL  Dimension: KZGOZO = 51
  Description:
  Ozone density elevation profile, i.e., ozone density at altitude levels of elevation grid. Values from 2 to 74 km are annual means from US Standard Atmosphere, 1976, for 45N (See NOAA, 1976) and values at 0 and 1 km are computed assuming a surface mixing ration of 40 ppb and standard atmosphere. Values are number density ($n_{O3}$ cm$^{-3}$). The total ozone column, assuming a linear behavior between elevation points, is 349.82 DU.
  Default: 1.02E+12, 9.2E+11, 6.8E+11, 5.8E+11, 5.7E+11, 6.5E+11, 1.13E+12, 2.02E+12, 2.35E+12, 2.95E+12, 4.04E+12, 4.77E+12, 4.86E+12, 4.54E+12, 4.03E+12, 3.24E+12, 2.52E+12, 2.03E+12, 1.58E+12, 1.22E+12, 8.73E+11, 6.07E+11, 3.98E+11, 2.74E+11, 1.69E+11, 1.03E+11, 6.64E+10, 3.84E+10, 2.55E+10, 1.61E+10, 1.12E+10, 7.33E+09, 4.81E+09, 3.17E+09, 1.72E+09, 7.5E+08, 5.4E+08, 2.2E+08, 1.7E+08, 12*0

• Card key word: O3_H
  Code variable: OZOHSC  Type: REAL  Dimension: 1
  Description:
  Ozone density scale height.
  Default: 4.50E+05
5.1.1.10 SO₂ molecule elevation profile definition:

The following variables are used in SUBROUTINE SETSO2. The SO₂ molecular density is given for layers of the atmosphere where the density is assumed to be constant. Note: the treatment of SO₂ and NO₂ molecules is identical.

- Card key word: SO₂T
  Code variable: SO2TIT  Type: CHARACTER*32  Dimension: 1
  Description:
  Title for the SO₂ molecule elevation profile definition. This variable does not influence the radiation transfer simulation and is only used when input values are printed out.
  Default: ‘Default SO₂ 1 ppb < 1km, 0 above’

- Card key word: SO₂N
  Code variable: SO2NEW  Type: REAL  Dimension: 1
  Description:
  Overhead SO₂ total column (in molec cm⁻²). An SO₂ molecular density elevation profile can be defined (see below). For every cell of the SO₂ elevation profile, the SO₂ molecular density is converted to a “partial SO₂ column”, and the default SO₂ total column is obtained by summing all the SO₂ partial columns. If a new SO₂ total column is given, the SO₂ molecular density profile is adjusted (scaled) such that the SO₂ total column is equal to SO2NEW. When SO2NEW is less than zero, the SO₂ column is determined by summing the SO₂ partial columns.
  Note: Because the SO₂ elevation grid is interpolated on the general elevation grid, it matters whether the scaling is performed before or after the interpolation (see section 5.1.1.4). The scaling is done on the profile on the general grid after interpolation, and not on the original SO₂ molecule profile before interpolation.
  Default: -999.0  i.e., SO₂ column defined by SO₂ density elevation profile.

- Card key word: SO₂Z
  Code variable: NBSO2Z  Type: INTEGER  Dimension: 1
  Description:
  Number of elevation cell boundaries (i.e., the number of elevation cells plus 1) for the SO₂ elevation profile.
  Default: 3
• Card key word: SO2G
  Code variable: ZG_SO2  Type: REAL  Dimension: KZGSO2 = 11
  Description:
  SO2 molecular density elevation profile grid, i.e., levels at which the SO2 molecular
density is given. The SO2 molecular density elevation grid must be in a strictly
ascending order. The elevation grid is given in kilometer.
  Default: 0, 1, 1.000001, 8*0

• Card key word: SO2D
  Code variable: SO2DEF  Type: REAL  Dimension: KZGSO2 = 11
  Description:
  SO2 molecular density elevation profile, i.e., SO2 molecular density at altitude
levels of elevation grid. Values are number density ($n_{SO2} \text{ cm}^{-3}$). The default values
 correspond to 1 ppb of SO2 molecule in the lowest kilometer of the atmosphere and
zero above. Given the standard atmosphere, 1 ppb is equivalent to $2.69 \cdot 10^{10} \text{ cm}^3$.
  Default: 2.69E10, 2.69E10, 0, 8*0

• Card key word: SO2H
  Code variable: SO2HSC  Type: REAL  Dimension: 1
  Description:
  SO2 molecular density scale height.
  Default: 4.50E+05

5.1.1.11 NO2 molecule elevation profile definition:

The following variables are used in SUBROUTINE SETNO2. The NO2 molecular density is
given for layers of the atmosphere where the density is assumed to be constant. Note: the
treatment of SO2 and NO2 molecules is identical.

• Card key word: NO2T
  Code variable: NO2TIT  Type: CHARACTER*32  Dimension: 1
  Description:
  Title for the NO2 molecule elevation profile definition. This variable does not
influence the radiation transfer simulation and is only used when input values are
printed out.
  Default: ‘Default NO2 1 ppb < 1km, 0 above’
Card key word: **NO2N**

**Code variable:** NO2NEW  **Type:** REAL  **Dimension:** 1

**Description:**
Overhead NO\(_2\) total column (in molec cm\(^{-2}\)). An NO\(_2\) molecular density elevation profile can be defined (see below). For every cell of the NO\(_2\) elevation profile, the NO\(_2\) molecular density is converted to a “partial NO\(_2\) column”, and the default NO\(_2\) total column is obtained by summing all the NO\(_2\) partial columns. If a new NO\(_2\) total column is given, the NO\(_2\) molecular density profile is adjusted (scaled) such that the NO\(_2\) total column is equal to NO2NEW. When NO2NEW is less than zero, the NO\(_2\) column is determined by summing the NO\(_2\) partial columns.

**Note:** Because the NO\(_2\) elevation grid is interpolated on the general elevation grid, it matters whether the scaling is performed before or after the interpolation (see section 5.1.1.4). The scaling is done on the profile on the general grid after interpolation, and not on the original NO\(_2\) molecule profile before interpolation.

**Default:** -999.0  i.e., NO\(_2\) column defined by NO\(_2\) density elevation profile.

Card key word: **NO2Z**

**Code variable:** NBNO2Z  **Type:** INTEGER  **Dimension:** 1

**Description:**
Number of elevation cell boundaries (i.e., the number of elevation cells plus 1) for the NO\(_2\) elevation profile.

**Default:** 3

Card key word: **NO2G**

**Code variable:** ZG_NO2  **Type:** REAL  **Dimension:** KZGNO2 = 11

**Description:**
NO\(_2\) molecular density elevation profile grid, i.e., levels at which the NO\(_2\) molecular density is given. The NO\(_2\) molecular density elevation grid must be in a strictly ascending order. The elevation grid is given in kilometer.

**Default:** 0, 1, 1.000001, 8*0
• Card key word: NO2D
  Code variable: NO2DEF  Type: REAL  Dimension: KZGNO2 = 11
  Description:
  NO₂ molecular density elevation profile, i.e., NO₂ molecular density at altitude levels of elevation grid. Values are number density \( n_{\text{NO}_2} \text{ cm}^{-3} \). The default values correspond to 1 ppb of NO₂ molecule in the lowest kilometer of the atmosphere and zero above. Given the standard atmosphere, 1 ppb is equivalent to \( 2.69 \times 10^{10} \text{ cm}^{-3} \).
  Default: 2.69E10, 2.69E10, 0, 8*0

• Card key word: NO2H
  Code variable: NO2HSC  Type: REAL  Dimension: 1
  Description:
  NO₂ molecular density scale height.
  Default: 4.50E+05

5.2 Command description and syntax

A command is a line of the input file starting with a command key word in the four first columns of the line. While the card syntax is the same for every card, the command syntax depends on the command. In general, the command consists of the command key word followed by values and possibly other key words.

While cards are used to set code variable values with one to one correspondence between card key words and code variables, commands are used to request TUVAQM to perform some actions such as reading files, for example.

Example

```
! Scales aerosols total optical depth by +/- 20%
SCAN AODD(:51) 4 0.80 1.20 REL
```

The command key word is SCAN that is used to indicate which parameter is varied for sensitivity analysis. SCAN is followed by a card key word such as AODD. Other values and key words follow.

Commands can be omitted. When commands are not given, default actions will be executed. The SCAN command has an influence only when using executable s_tuv… for sensitivity analysis. When using other programs or an AQM (see section 7), the SCAN command has no effect.
5.2.1 Command list

5.2.1.1 End-of-file command

- Command key word: END
  Command syntax: END
  Description:
  When the END command is encountered, TUVAQM initialization routine stops to read the general cards and commands input file. Whatever is after the END command is not read.

5.2.1.2 Sensitivity analysis SCAN command

A test program is provided with TUVAQM for sensitivity analysis (see section 7). Sensitivity analysis is performed by varying parameters in a given number of steps between given limits. Several parameters can be varied simultaneously (up to five). When more than one parameter is varied, the sensitivity analysis program systematically computes a radiation transfer simulation for all combination of the varying parameters. The parameters that can be varied are the parameters that can be set with the card key words. In case an AQM program or a test program other than the sensitivity analysis program is used, the SCAN command has no effect.

- Command key word: SCAN
  Command syntax: SCAN Card # of steps Limits Key words
  Description:
  SCAN is used for repeating radiation transfer simulation with one ore more parameter varied in a systematic way. The parameter to be varied is defined by the card key word given after the SCAN command. The card key word refers to one of the key words described in section 5.1. (Note: Only a limited set of the variables linked to the card key words can be varied with SCAN. These are of type INTEGER, REAL or DOUBLE PRECISION, and are not part of the general parameters or grid parameters indicated in section 5.1). When the parameters are linked to variable arrays, it is possible to define the range of indices that are affected by the SCAN command, e.g., AOMD(13:25) means the indices 13 to 25 for the variable linked to the card AOMD. If no index range is given and the variable is an array, the SCAN command applies to the whole range of indices. The parameter is varied between two limits in a certain number of steps. The program

33
uses one more value than there are steps for each parameter. For example, the
program makes four steps between five values.

Key words can be added to affect the way the parameter values are varied:

- **REL**: The limits indicate a relative value with respect to the nominal
  value of the parameter.
- **DIF**: The limits indicate a difference with respect to the nominal value
  of the parameter.
- **BND**: When SCAN commands for more than one parameter include the
  BND key word, these parameters are varied together. Specifically, the
  same number of steps are taken for these parameter, and a simulation is
  performed with all parameters at their first value, then another with all
  parameters at their second value, etc. (Normally, the program varies the
  parameters one by one trying all combinations of values.)

**Example**:

```plaintext
!
! Scan ozone total column from -50% to +50% of 300 DU
! in 10 steps.
! First set nominal value to 300 DU
O3_N 300
SCAN O3_N 10 -0.5 0.5 REL DIF
!
! Scan aerosols single scattering albedo (range 1:50)
! from 0.8 to 1.0 in 4 steps.
SCAN AOMD(:50) 4 0.80 1.00
```

### 5.2.1.3 Atmospheric light-absorbing gas cross-section definition

Light absorption by atmospheric gases is simulated in **TUVAQM**. The wavelength-dependent
absorption cross section is read in data files. Some default data files are provided. The command
**XSEC** allows using other files for defining the absorption cross-section for the gases.

- **Command key word**: XSEC
- **Command syntax**: XSEC  File #  Subcommand  Values
- **Description**:

  XSEC is used to define which files contain cross section data for the light-
  absorbing gas in the atmosphere, and the way to read them. Every XSEC command
refers to a file identified by the file number (second element of the XSEC command line). The command allows defining the name of the file as well as the corresponding absorbing gas. It is assumed that the data are organized in columns with one or two wavelength columns and one or more columns for the corresponding cross sections. It is possible to indicate which columns contain the wavelengths and cross sections. When cross sections are in more than one column, each column is valid for a temperature that is also defined with the XSEC command. More than one file can refer to the same light-absorbing gas. In this case, each file is usually valid for a given wavelength range that is also indicated with XSEC.

Subcommand:

- **Subcommand key word:** FNAME
  
  **Subcommand syntax:** XSEC File # FNAME Gas Filename

  **Description:**
  
  XSEC FNAME is used to define the file name and light-absorbing gas corresponding to a given file number. The gas name is a key word that needs to be recognized. The gas name key words are O2, O3, SO2 and NO2.

  **Example:**
  
  XSEC 2 FNAME O3 DATAE1/O3/O3.molina.abs

  **Note:** The command XSEC File # FNAME can be used without arguments. In this case, what is kept in memory for the given file number is reset, and no file is read. When the user wants to use less files than the default, it is necessary to reset the unused file number that are active by default.

- **Subcommand key word:** WLLIM
  
  **Subcommand syntax:** XSEC File # WLLIM Limits Limit type

  **Description:**
  
  When more than one file contains data, it is assumed the files are valid for different wavelength ranges. For example, the cross sections may be measured at different temperature intervals for different wavelength ranges. XSEC WLLIM is used to define what is the wavelength range of validity for a given file. The limits are two wavelengths (nm), and the limit type is a FORTRAN logical expression defining how the limits are used. There are 8 recognized limit types:
The limit type should be strictly identical to one of the recognized types; however, blank characters do not count and can be included anywhere in the limit type.

Example:
XSEC 2 WLLIM 240.5 347.0 a .lt. wl .and. wl .lt. b

* Subcommand key word: WLCOL
Subcommand syntax: XSEC File # WLCOL Column 1 Column 2
Description:
XSEC WLCOL is used to define what columns in the given file contain the wavelength data. When only one column is indicated, it is assumed that the corresponding cross section is given at that wavelength. When two columns are indicated, they are assumed to be the limits of a bin for which the corresponding cross-section is given.

Example:
XSEC 2 WLCOL 2 3

* Subcommand key word: XSCOL
Subcommand syntax: XSEC File # XSCOL Col1 Temp1 ...
Description:
XSEC XSCOL is used to define what columns in the given file contain the cross section data. When only one column is indicated, the cross section is assumed not to depend on temperature. When more columns are indicated, they are assumed to be valid at a given temperature each, and every column number should be followed by the corresponding temperature.

Example:
XSEC 2 XSCOL 2 226 3 263 4 298
5.2.1.4 Extraterrestrial solar flux definition

The intensity of the extraterrestrial solar flux is the primary input in TUVAQM. The extraterrestrial solar flux spectrum is read in data files. Default data files are provided. The command FLUX allows using other files for defining the extraterrestrial solar flux spectrum.

- Command key word: FLUX
  Command syntax: FLUX File # Subcommand Values
  Description:
  FLUX is used to define which file contains the extraterrestrial solar flux spectrum data, and the way to read it. The command allows defining the name of the file and its syntax. It is assumed that the data are organized in columns with one or two columns for wavelength and one or more for the corresponding extraterrestrial solar flux. It is possible to indicate which columns contain the wavelengths and extraterrestrial solar flux. More than one file can be used to define the extraterrestrial solar flux spectrum. In this case, each file is usually valid for a given wavelength range that is also indicated with FLUX.

- Subcommand key word: FNAME
  Subcommand syntax: FLUX File # FNAME Filename
  Description:
  FLUX FNAME is used to define the file name corresponding to a given file number.

Example:

FLUX 1 FNAME DATA1/SUN/susim_hi.flx

Note: The command FLUX File # FNAME can be used without arguments. In this case, what is kept in memory for the given file number is reset, and no file is read. When the user wants to use less files than the default, it is necessary to reset the unused file numbers that are active by default.

- Subcommand key word: WLLIM
  Subcommand syntax: FLUX File # WLLIM Limits Limit type
  Description:
  When more than one file contains data, it is assumed the files are valid for different wavelength ranges. For example, the solar flux measurements can come from different references for different wavelength ranges. FLUX
WLLIM is used to define what is the wavelength range of validity for a given file. The limits are two wavelengths (nm), and the limit type is a FORTRAN logical expression defining how the limits are used. There are 8 recognized limit types:

• a.lt.wl.and.wl.lt.b
• a.le.wl.and.wl.lt.b
• a.lt.wl.and.wl.le.b
• a.le.wl.and.wl.le.b
• wl.lt.a.or.b.lt.wl
• wl.le.a.or.b.lt.wl
• wl.lt.a.or.b.le.wl
• wl.le.a.or.b.le.wl

The limit type should be strictly identical to one of the recognized types; however, blank characters do not count and can be included anywhere in the limit type.

Example:

```
FLUX 1 WLLIM 0 350 a .lt. wl .and. wl .le. b
```

- **Subcommand key word:** WLCOL
- **Subcommand syntax:** FLUX File # WLCOL Column 1 Column 2
- **Description:**

  FLUX WLCOL is used to define what columns in the given file contain the wavelength data. When only one column is indicated, it is assumed that the corresponding solar flux is given at that wavelength. When two columns are indicated, they are assumed to be the limits of a bin for which the corresponding solar flux is given.

Example:

```
FLUX 1 WLCOL 1
```

- **Subcommand key word:** WLMUL
- **Subcommand syntax:** FLUX File # WLMUL Factor
- **Description:**

  FLUX WLMUL defines a wavelength factor for use when the wavelengths are not given in the library routine units (nm). The multiplicative factor is to be applied to all wavelength input from the given file.
Example (units are microns):

```
FLUX 3 WLMUL 1000.0
```

- **Subcommand key word:** FXCOL
  
  **Subcommand syntax:** `FLUX File # FXCOL Col1 Temp1 ...`

  **Description:**
  
  FLUX FXCOL is used to define what columns in the given file contain the cross section data. When only one column is indicated, it is assumed that the solar flux is given at the wavelength corresponding to the current line. When more columns are indicated, it is assumed that each column gives flux values at a wavelength with a given offset from the base wavelength given for each line (a different offset for each column allows reducing the number of lines to be read for high resolution files). Every column number should be followed by the corresponding offset.

  Example:

  ```
  FLUX 3 FXCOL 2 .00 3 .05 4 .10 5 .15 6 .20 7 .25
  ```

- **Subcommand key word:** FXMUL
  
  **Subcommand syntax:** `FLUX File # FXMUL Factor Flag`

  **Description:**
  
  FLUX FXMUL defines a solar flux factor for use when the file solar flux is not given in the library routine units (watts m$^{-2}$ nm$^{-1}$). The multiplicative factor is to be applied to all solar flux inputs from the given file. A logical flag can be given to indicate when the solar flux is given as energy quanta. In this case, to obtain the solar flux in the working units, TUVAQM divides the flux by the wavelength.

  Example (solar flux given as light quanta):

  ```
  FLUX 2 FXMUL 1.985E-12 .TRUE.
  ```

- **Subcommand key word:** INTSC
  
  **Subcommand syntax:** `FLUX File # INTSC Interp. scheme`

  **Description:**
  
  FLUX INTSC allows choosing the scheme to be used to interpolate the solar flux data from the wavelength grid indicated in the file to the general wavelength grid. Three interpolation schemes are possible:
1) The original input data are given on single, discrete points of an arbitrary grid and are linearly interpolated onto the specified discrete target grid. A typical example would be the re-gridding of a given data set for the vertical temperature profile to match the specified altitude grid.

2) The original input data are given on single, discrete points of an arbitrary grid and are linearly interpolated onto a specified set of target bins. In general, this is the case for most of the weighting functions (action spectra, molecular cross section, and quantum yield data) which have to be matched onto the specified wavelength intervals. The average value in each target bin is found by averaging the trapezoidal area underneath the input data curve (constructed by linearly connecting the discrete input values).

3) The input data are given on a set of bins representing the input quantity integrated over the range of each bin (the area of each bin) and are matched onto another set of bins (target grid). The resulting area in a given bin of the target grid is calculated by simply adding all fractional areas of the input data that cover that particular target bin.

Example (solar flux given as light quanta):

```
FLUX 1 INTSC 2
```

### 5.2.1.5 Photolysis reaction cross section and quantum yield definition

**TUVAQM** computes the solar irradiance in the atmosphere. It can be used to compute the photolysis reaction rate coefficient (in sec\(^{-1}\)) for given reactions, providing the corresponding cross-sections and quantum yields are known. The cross sections and quantum yields are read in data files. Currently, **TUVAQM** reads cross-sections and quantum yields for 20 photolysis reactions defined in SAPRC97 ([http://cert.ucr.edu/~carter/saprc97.htm](http://cert.ucr.edu/~carter/saprc97.htm)). The SAPRC97 reactions are:

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Cross-section</th>
<th>Quantum Yield</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (\mathrm{NO}_2)</td>
<td>(\mathrm{NO}_2 + \text{hv} \Rightarrow \mathrm{NO} + \text{O})</td>
<td></td>
</tr>
</tbody>
</table>
6  17  HONO  HONO + hv  ⇒  HO. + NO
7   31  H₂O₂  H₂O₂ + hv  ⇒  2 HO.
8   B7  CO₂H  -OOH + hv  ⇒  HO₂. + HO.
9   C1  HCHO  HCHO + hv  ⇒  2 HO₂. + CO
10  C2  HCHO  HCHO + hv  ⇒  H₂ + CO
11  C11A CCHO  CCHO + hv  ⇒  CO + HO₂. + HCHO + RO₂⁻R. + RO₂.
12  C26 RCHO  RCHO + hv  ⇒  CCHO + RO₂⁻R. + RO₂. + CO + HO₂.
13  C39  Acetone  ACET + hv  ⇒  CCO-O₂. + HCHO + RO₂⁻R. + RCO₃. + RO₂.
14  C57  MEK  MEK + hv + 0.1  ⇒  CCO-O₂. + CCHO + RO₂⁻R. + RCO₃. + RO₂.
15  C58A Glyoxal  GLY + hv  ⇒  0.8 HO₂. + 0.45 HCHO + 1.55 CO
16  C58B Glyoxal  GLY + hv + 0.029  ⇒  0.13 HCHO + 1.87 CO
17  C68A Methyl-Glyoxal  MGLY + hv  ⇒  HO₂. + CO + CCO-O₂. + RCO₃.
18  C68B Methyl-Glyoxal  MGLY + hv + 0.107  ⇒  HO₂. + CO + CCO-O₂. + RCO₃.
19  G31  Benzaldehyde  BALD + hv + 0.05  ⇒  7 -C
20  G8  Acrolein  AFG1 + hv + 0.077  ⇒  HO₂. + HCOCO-O₂. + RCO₃.
21  U2HV Acrolein  AFG2 + hv  ⇒  HO₂. + CO + CCO-O₂. + RCO₃.

Note: reaction 20 (G8) and 21 (U2HV) use the same cross section and quantum yield definition and only the rate coefficient for reaction 20 is computed.

No command is installed yet to allow choosing the reactions. Code changes are needed to implement a different set of reactions (see section 8).
6 Region and time-dependent card description

The region and time-dependent card input file indicates the value of the parameters assumed to vary with time and space (see section 3). These parameters correspond to parameters defined with cards in section 5 (latter referred to as general cards). Within an AQM, the library routines use the values read in the region and time-dependent card input files to substitute for the default values (or the values set with general cards) with the appropriate values for the current region and time interval. A list of the parameters that are modified and the corresponding general cards is given below:

- **zenith angle**: LTIM, NSTO, SPNT | LTIM is set to .FALSE., NSTO is set to the number of zenith angles requested for the region and time interval, and SPNT is set to the zenith angle values.
- **elevation grid**: NBZG, ZGRD | NBZG is set to the number of elevation levels requested for the region and time interval, and ZGRD is set to the elevation level values.
- **ground albedo**: ALNW, ALWG, ALBD | ALNW is set to 1, ALWG is set to one wavelength bin covering the whole wavelength range, and ALBD is set to the ground albedo value requested for the region.
- **total ozone column**: O3_N | O3_N is set to the total ozone column requested for the region and time interval.
- **total aerosol optical depth**: AERF, AODN | AERF is set to .TRUE., and AODN is set to the total aerosol optical depth requested for the region and time interval.
- **aerosol single scattering albedo**: AOMD | AOMD is set to the single scattering albedo requested for the region and time interval.
- **aerosol asymmetry factor**: AGDE | AGDE is set to the asymmetry factor requested for the region and time interval.

The syntax of the region and time-dependent card input file is similar to the syntax of general card and command input files. However, there are more restrictions: no elements of the file can be skipped, and the elements should appear in a definite order. Because of these differences, to avoid confusion, the key words in the region and time-dependent card input file are different from
the corresponding key words of the general card and command input file. All inputs in the file take the form of one key word followed by the corresponding values, possibly on several lines, until the next key word. Values are separated by one or more blank characters or tabulation characters. In order to increase the readability of the input file, comment lines starting with an exclamation mark can be included at any time. The sequence of information in the region and time-dependent card input file is the following:

1. **Number of regions and time interval limits**
   
   When using **TUVAQM** within an AQM, the user must divide the modeling domain in regions and the modeling episode in time intervals such that the atmospheric properties are relatively constant in a given region and time interval. The number of regions and time interval limits is given here. Note: since the time interval limits are given here, there should be one more limit than there are time intervals.

   **Key word:** NBRT  
   **Syntax:** NBRT Number of regions Number of time limits  
   **Example:** NBRT 2 4

2. **Time interval limits**
   
   The time interval limits are given in the AQM working time units (usually in minutes). The limits measure the simulated time elapsed since the beginning of the simulation. They should be in a strictly ascending order. There should be as many limits as declared with NBRT.

   **Key word:** TINT  
   **Syntax:** TINT Limit 1 Limit 2 Limit 3...  
   **Example:** TINT 0 240 480 720

3. **Number of ground elevations per region**
   
   In each region, the ground elevation may vary substantially. When used within an AQM, **TUVAQM** computes simulations for a certain number of ground elevations that are representative for the region. The results for a given AQM cell with a given ground elevation are interpolated from the results of the simulations with the representative ground elevations. The number of representative ground elevations for each region is set with NGEL. There should be as many numbers as there are regions (defined with NBRT). The interpolation is done with a 3-points 2nd-order polynomial algorithm; thus, there should be at least 3 representative ground elevations per region.

   **Key word:** NGEL  
   **Syntax:** NGEL # of elevation 1 # of elevation 2 # of elevation 3...  
   **Example:** NGEL 4 3
4. **Ground elevations in each region**

The representative ground elevations in each region are given in the units of the general elevation grid, that is kilometers above sea level (*not meters above sea level*). GREL is the key word indicating the ground elevation input. It should be repeated as many time as there are regions, and for each region there should be as many inputs as declared with NGEL. Each series of ground elevations should be in a strictly ascending order.

**Key word:** GREL  
**Syntax:** GREL Elevation 1 Elevation 2 Elevation 3...  
**Example:**  
! Region 1  
GREL 0.0 0.5 1.0 1.5  
! Region 2  
GREL 0.2 0.6 1.0

5. **Number of zenith angle steps per region and time interval**

The solar zenith angle depends on the location, date and time. Since time intervals can be defined as lasting several hours, the solar zenith angle may change significantly within a time interval. In case the modeling domain is very large, the zenith angle range for a given time interval may also change from one region to the other. Simulations are computed for zenith angles representative for each region and time interval. The results at a given time for a given AQM cell resulting in a given zenith angle are interpolated from the results of the simulations with the representative zenith angles. The number of representative zenith angles for each region and time interval is set with NBZN. It should be repeated as many times as there are regions, and for each region there should be as many inputs as there are time intervals. The interpolation is done with a 3-points 2nd-order polynomial algorithm; thus, there should be at least 3 representative zenith angles per region and time interval.

**Key word:** NBZN  
**Syntax:** NBZN # of zenith angles 1 # of zenith angles 2 # of zenith angles 3...  
**Example:**  
! Region 1 (3 time intervals)  
NBZN 8 5 8  
! Region 2 (3 time intervals)  
NBZN 8 4 8
6. **Zenith angle steps in each region and time interval**

   The representative solar zenith angle steps in each region and time interval are given in degrees. **ZNST** is the key word indicating the zenith angle input. It should be repeated as many times as there are regions multiplied by time intervals, and for each region and time interval there should be as many inputs as declared with **NBZN**. Each series of zenith angles should be in a strictly ascending order *even though the solar zenith angle decreases during the afternoon.*

   **Key word:** **ZNST**

   **Syntax:**

   ZNST Zenith angle 1 Zenith angle 2 Zenith angle 3...

   **Example:**

   ```
   ! Region 1 (3 time intervals with 8, 5 and 8 steps)
   ZNST 40 50 60 70 75 80 85 90
   ZNST 0 10 20 30 40
   ZNST 40 50 60 70 75 80 85 90
   !
   ! Region 2 (3 time intervals with 8, 4 and 8 steps)
   ZNST 40 50 60 70 75 80 85 90
   ZNST 10 20 30 40
   ZNST 40 50 60 70 75 80 85 90
   ```

7. **Number of levels in elevation grid per region and time interval**

   **TUVAQM** uses one general elevation grid for performing the radiation transfer calculations. This grid is user-defined. All quantities that depend on elevation are given a fixed value in each cell of the general elevation grid. The choice of the elevation grid should depend on the altitude resolution of the input data (air pressure, aerosol and absorbing gas density, etc.) and the expected resolution of the output. The different inputs are all given on their own elevation grid, and library routines map them on the general grid. Thus, having a grid with smaller cells than the cells of the finest input grid does not improve the precision of the calculation. Similarly, the AQM callable routine of **TUVAQM** maps the results computed on the general grid to the desired output grid. Thus, a general grid with cells larger than the output grid will lead to a loss of resolution. For each region and time interval a specific elevation grid is defined. The number of levels in the elevation grid for each region and time interval is set with **NBHL**. It should be repeated as many times as there are regions, and for each region there should be as many inputs as there are time intervals.
Key word: NBHL

Syntax:  

NBHL  # of altitude levels 1  # of altitude levels 2  
# of altitude levels 3...

Example:  

! Region 1 (3 time intervals)  
NBHL 34 32 30

! Region 2 (3 time intervals)  
NBHL 32 32 31

8. Elevation grid levels in each region and time interval

The levels of the general elevation grid are given with the key word HLEV. Since different ground elevations can be defined for the same region, the elevation grid is defined relative to the ground elevation level, and not from the sea level. The levels are given in kilometers above ground level. The bottom of the lowest cell is always assumed to be the ground level. Thus the first elevation level is the top of the first cell. Consequently, the number of elevation levels (see NBHL) is equal to the number of cells in the elevation grid. The HLEV key word should be repeated as many times as there are regions multiplied by time intervals, and for each region and time interval there should be as many inputs as declared with NBHL. Each series of elevations should be in a strictly ascending order.

Key word: HLEV

Syntax:  

HLEV Elevation 1   Elevation 2   Elevation 3...

Example:  

! Defines elevation levels in each region and 
! time interval

! Region 1 (3 time intervals with 34, 32 & 30 levels)
HLEV 0.1 0.2 0.3 0.4 0.5 1.0 2.0 3.0 4.0 5.0 8.0 12.0  
25. 26. 27. 28. 29. 30. 35. 40. 45. 50.

HLEV 0.1 0.3 0.5 1.0 2.0 3.0 4.0 5.0 8.0 12.0  
25. 26. 27. 28. 29. 30. 35. 40. 45. 50.

HLEV 0.1 0.3 0.5 1.0 2.0 3.0 5.0 12.0  
25. 26. 27. 28. 29. 30. 35. 40. 45. 50.
9. **Ground albedo in each region**

An average ground albedo (a single value valid at all wavelengths) can be defined for each region. The `GALB` key word should be given with as many inputs as there are regions.

**Key word:** GALB

**Syntax:**

```plaintext
GALB Ground albedo 1 Ground albedo 2 Ground albedo 3...
```

**Example:**

```plaintext
! Average ground albedo in each region
GALB 0.08 0.20
```

10. **Total ozone column in each region and time interval**

The total ozone column is one of the important atmospheric properties that should be defined separately for each region and time interval. The total ozone column is given in Dobson units with the `O3TC` key word. It should be repeated as many times as there are regions, and for each region there should be as many inputs as there are time intervals.

**Key word:** O3TC

**Syntax:**

```plaintext
O3TC O3 column 1 O3 column 2 O3 column 3...
```

**Example:**

```plaintext
! Gives O3 total ozone column in each region and time interval
!
! Region 1 (3 time intervals)
O3TC 305.0 300.3 302.5
```
11. Aerosol properties in each region and time interval

Average atmospheric aerosol properties that should be defined separately for each region and time interval are the aerosol total optical depth, the aerosol single scattering albedo and the aerosol asymmetry factor. The aerosol total optical depth is given in form of attenuation factor per kilometer of atmosphere. It is used to scale the atmospheric aerosol optical depth profile (either the default one or the one defined as discussed in section 5). The single scattering albedo and asymmetry factor are average quantities that apply to all altitude cells. (It was assumed that data on their elevation profile were too scarce to warrant a more precise definition). The key word for total optical depth is ATOD, AOME for the single scattering albedo, and AGDE for the asymmetry factor. These key words should be given one after the other as many times as there are regions, and should each have as many inputs as there are time intervals.

Key word: ATOD, AOME, AGDE

Syntax:
- ATOD Optical depth 1 Optical depth 2 Optical depth 3...
- AOME S.s. albedo 1 S.s. albedo 2 S.s. albedo 3...
- AGDE Asym. fact. 1 Asym. fact. 2 Asym. fact. 3...

Example:

! Aerosol properties in each region and time interval
!
! Region 1 (3 properties for 3 time intervals)
ATOD  0.5  1.0  0.7
AOME  0.90 0.80 0.85
AGDE  0.61 0.66 0.61
!
! Region 2 (3 properties for 3 time intervals)
ATOD  0.1  0.3  0.3
AOME  0.90 0.90 0.90
AGDE  0.61 0.61 0.61

7 Installing TUVAQM

The library routines are written in FORTRAN 77. The library routine package is distributed as an archive file created with the UNIX tar utility and compressed with UNIX gzip. On most UNIX system, the archived compressed file can be uncompressed with the command

```bash
48
```
gunzip tuvaqm.tar.zip, and de-archived with the command tar -xf tuvaqm.tar. At the upper level, the archive file contains two UNIX directories and a Readme file describing the directories. The first directory called Dcode contains the FORTRAN code for the library routines. The second directory called Drun contains supporting files needed for most radiation transfer simulations. When the code is compiled and an executable created, it should be run within Drun, because supporting file names are given as relative paths and not absolute paths. It is also possible to create an object library containing the main AQM-callable routine (subroutine tuvaqm) and the other library routines. Any executable calling tuvaqm will need the supporting data within Drun.

With the code is included a makefile for the UNIX make utility. The makefile includes several targets (entry points). The main target (tuvlib.a) is used for creating the object library containing the main AQM-callable routine and the library routines. Hence the command make or make tuvlib.a issued within the directory Dcode will create the object library tuvlib.a that can be linked with an AQM program calling tuvaqm. The other targets are used to create standalone executables using the library routines. The name of the executable created is identical to the target. For example, the command make t_taqm will create the executable t_taqm. Theses executables differ in the main program used, and the output package used. t_taqm and t_taqmh use a main program that tests the library routines with a calling sequence similar to what an AQM would use. t_tuvn, t_tuvh and t_tuvw use a main program that only performs one radiation transfer simulation. s_tuvn, s_tuvh and s_tuvw use a main program that allows rerunning the radiation transfer model while varying desired input parameters. t_tuvn and s_tuvn do not save results, whereas t_tuvh and s_tuvh save photolysis reaction rate coefficient results in a format described as CERN RZ libraries (http://wwwinfo.cern.ch/asdoc/, the CERN libraries are needed only for these two executables). t_tuvw and s_tuvw save the results with simple FORTRAN write statements.

The makefile uses the standard FORTRAN compiler (variable FC). On most UNIX operating systems this is defined as f77. However, it may not be available on some system, and FC may not be defined appropriately. Such a problem can occur on LINUX systems where one can try to define FC = g77, or on systems featuring only the newer f90 where defining FC = f90 should solve the problem.

Use of TUVAQM can be computationally intensive, especially when it is called repeatedly. To lower the computational burden, it is a good idea to use the code optimization utility available with most compilers. The original version of the makefile uses compilation flags (variable FFLAGS) that produce aggressive optimization on a SUN Sparc station without use of
multiprocessing. These flags usually depend on the compiler and the operating system, and it is up to the user to find the appropriate flags.

In the directory Drun, some input and output files are provided so that the user can check that the desired executables are correctly installed. A Readme file in the directory describes its content.

8 Code description

SUBROUTINE TUVAQM (in file tuvaqm.f) is the only routine that should be called from an Air Quality Model. It uses seven generic routines that perform a group of related tasks. The seven generic routines call numerous other routines that perform a single task each. While the seven generic routines are each contained in separate files, the other supporting routines are grouped into files according to their purpose. In addition there are four test main programs.

Beside TUVAQM, the seven generic routines are DEP_RD, GETPAR, GTPLOC, MODPAR, TUV_RD, TUV_RN and ZG_MOD (files dep_rd.f, getpar.f, gtploc.f, modpar.f, tuv_rd.f, tuv_rn.f and zg_mod.f, respectively). All these routines are thoroughly documented.

8.1.1 Generic routines

8.1.1.1 tuvaqm.f

TUVAQM is a general main routine for using TUV within an AQM program. TUVAQM should be called once at initialization time, and then it should be called every time J values are needed for the photolysis reactions. TUVAQM provides the J-values for all reactions defined at the initialization step, for all cells in a column defined in the calling sequence.

Content:

SUBROUTINE TUVAQM: Radiative transfer initialization or computation.

8.1.1.2 dep_rd.f

DEP_RD is a routine that reads the user’s defined radiative transfer atmospheric properties for each AQM time interval and region.

Content:

SUBROUTINE DEP_RD: Reads time interval and region definition.
8.1.1.3 getpar.f

GETPAR is a routine that gets the value of a parameter (e.g. total ozone column) or an array of parameters whose position in the general parameter array has previously been identify by subroutine GTPLOC.

Content:
SUBROUTINE GETPAR: Retrieves the value of parameters identified by gtploc.

8.1.1.4 gtploc.f

GTPLOC is a routine that returns the location of a given parameter (e.g. total ozone column) in the parameter array based on the datacard name linked to the parameter (in the case of total ozone column, the datacard name is O3_N). GTPLOC should be called at the beginning of the program once for each parameter that will need to be updated during execution.

Content:
SUBROUTINE GTPLOC: Finds the variable pertaining to a given input parameter.

8.1.1.5 modpar.f

MODPAR is a routine that modifies the value of a parameter (or a parameter array). The parameter is identified by the location returned from GTPLOC. MODPAR should be called before the routine computing the J-values (TUV_RN), once for each parameter modified since the last execution of TUV_RN.

Content:
SUBROUTINE MODPAR: Modify an input parameter or an array of input parameters.

8.1.1.6 tuv_rd.f

TUV_RD is an initialization routine that reads user’s directives and various data files. It should be called once at the beginning of the program.

Content:
SUBROUTINE TUV_RD: Performs general initialization and reads data file.

8.1.1.7 tuv_rn.f

TUV_RN is the routine simulating the radiation transfer, and computing the J-values.

Content:
SUBROUTINE TUV_RN: Sets up model atmosphere and calls the radiation transfer routine.
8.1.1.8 zg_mod.f

ZG_MOD is a routine that modifies the bottom part of the altitude grid, so that the lower part of the TUVAQM general z-grid corresponds to the AQM z-grid. ZG_MOD should be called before the routine computing the J-values (TUV_RN), if the AQM z-grid was modified since the last execution of TUV_RN.

Content:

SUBROUTINE ZG_MOD: Modifies the altitude grid to incorporate new layers.

8.1.2 Test programs

8.1.2.1 mn_try.f

File mn_try.f contains a test program using the routines described above.

Content:

MAIN MN_TUV: Main program to test the subroutines described above.

8.1.2.2 mn_sca.f

File mn_sca.f contains a program that allows rerunning a radiation transfer simulation while changing the value of desired input parameters step by step.

Content:

MAIN MN_TUV: Main program used for TUVAQM with input parameter value scan.
SUBROUTINE SCAVAL: Runs TUV_RN repeatedly while changing input parameter values.

8.1.2.3 tstaqm.f

File tstaqm.f contains a test program that uses TUVAQM in a calling sequence that emulates the one that could be found in an AQM.

Content:

MAIN TSTAQM: Main program used for testing TUVAQM in AQM calling sequence.

8.1.2.4 tstaqmh.f

File tstaqmh.f is identical to tstaqm.f except that CERN libraries are used for saving the program outputs.

Content:

MAIN TSTAQM: Main program used for testing TUVAQM in AQM calling sequence.
8.1.3 Other routines

8.1.3.1 setsca.f

When the program in mn_sca.f is chosen, setsca is used to choose what parameters are varied, and what are the values at each step.

Content:

SUBROUTINE SETSCA: Set initial values for a scan of possible parameter values.

BLOCK DATA BDSCAN: Initializes default data for parameter value scanning.

8.1.3.2 caread.f

File caread.f contains routines used to set and show values for parameters describing how the radiative transfer simulation should proceed.

Content:

SUBROUTINE CARINI: Initializes the parameter value input package.

SUBROUTINE CAREAD: Reads and interprets the lines of the parameter value file.

SUBROUTINE CARINF: Inputs one line of the parameter value file.

SUBROUTINE CAINCA: Interprets values for simple parameter definitions.

SUBROUTINE CAINCO: Interprets command lines.

FUNCTION LENOCC: Finds the last non-blank character of a string.

FUNCTION LVCATS: Finds a character string in a character string array.

SUBROUTINE LVUPCA: Changes the case of a character string to uppercase.

SUBROUTINE CARSHS: Shows current status of user’s input cards

SUBROUTINE CARSH1: Shows current status of one of the user’s input cards

SUBROUTINE CARSHL: Writes a string of logical data

SUBROUTINE CARSHI: Writes a string of integer data

SUBROUTINE CARSHR: Writes a string of real data

SUBROUTINE CARSHD: Writes a string of double precision data

SUBROUTINE CARSHC: Writes character strings data

8.1.3.3 inidef.f

File inidef.f contains a group of BLOCK DATA that gives default values to the parameters set by the code in caread.f
Content:

BLOCK DATA CARDEF: Defines keywords used for initializing parameters.
BLOCK DATA CARVAL: Gives default values for user-defined parameters.
BLOCK DATA CMDDEF: Defines keywords for initialization commands.
BLOCK DATA BDFXRD: Defines subcommands for reading solar flux data files.
BLOCK DATA BDXSRD: Defines subcommands for reading cross-section data files.

8.1.3.4 setcnd.f

File setcnd.f contains routines that set-up the environment conditions for simulating the radiative transfer once the parameters describing the environment conditions have been read.

Content:

SUBROUTINE SETAER: Sets up an altitude profile for aerosol characteristics.
SUBROUTINE SETAIR: Sets up an altitude profile for air density.
SUBROUTINE SETALB: Sets up a wavelength profile for ground albedo.
SUBROUTINE SETCLD: Sets up an altitude profile for cloud characteristics.
SUBROUTINE SETNO2: Sets up an altitude profile for NO2 absorption optical depth.
SUBROUTINE SETO2: Sets up an altitude profile for O2 absorption optical depth.
FUNCTION FCHAP: Computes the Chapman function (O2 absorption optical depth).
SUBROUTINE SCHU: Computes equivalent O2 absorption x-section in SR bands.
SUBROUTINE SETOZO: Sets up an altitude profile for O3 absorption optical depth.
SUBROUTINE SETSO2: Sets up an altitude profile for SO2 absorption optical depth.
SUBROUTINE SETTMP: Sets up an altitude profile for temperature.

8.1.3.5 setphr.f

File setphr.f contains code to define weighting functions for the photolysis reactions of interest. The radiative transfer simulation computes the wavelength-dependent irradiance at a given time and location. To compute photolysis reaction rate coefficients, wavelength-dependent cross sections and quantum yields need to be known. The weighting functions are defined as the product of the cross section times the quantum yield.

Content:

SUBROUTINE RDPHOD: Reads the weighting functions information.
FUNCTION NBARG: Returns the number of arguments in a character string.
SUBROUTINE SETPHO: Computes the weighting functions for photolysis reactions.
8.1.3.6 setflx.f

File setflx.f contains code for inputing the extraterrestrial (solar) flux. The extraterrestrial flux (mainly from solar radiation) is the source of all the primary radiation. It is a wavelength-dependent quantity read in one or more data files.

Content:

SUBROUTINE RDFXFI: Reads extraterrestrial (solar) flux data files.
SUBROUTINE SETFXR: Sets characteristics for reading solar flux data files.
SUBROUTINE SFXFXC: Gets solar flux data file format for solar flux definition.
SUBROUTINE SFXFXM: Gets a multiplicative factor for solar flux definition.
SUBROUTINE SFXINS: Gets an interpolation scheme number for wavelengths.
SUBROUTINE SFXLIM: Gets wavelength validity domain for solar flux data files.
SUBROUTINE SFXNAM: Gets extraterrestrial (solar) flux file names.

8.1.3.7 setxsn.f

File setxsn.f contains the code to define the absorption cross section (x-section) for different gases. Gases in the atmosphere absorb radiation. The absorption cross sections for the main absorbing gases in the atmosphere are read in data files.

Content:

SUBROUTINE RDXSFI: Reads gas absorption x-section files.
SUBROUTINE SETXSR: Sets characteristics for reading gas absorption x-sections.
SUBROUTINE SXSLIM: Gets wavelength validity domain for x-section data files.
SUBROUTINE SXSNAM: Gets gas absorption x-section file names.
SUBROUTINE SXSWLC: Gets x-section data file format for wavelength definition.
SUBROUTINE SXSXSC: Gets x-section data file format for x-section definition.

8.1.3.8 tuvgrd.f

File tuvgrd.f contains the code to define the altitude and wavelength general grid. All quantities that depend on the altitude or the wavelength are defined on grids. The user can define the grids. Most radiation quantities are wavelength dependent. Quantities that depend on temperature or pressure are altitude (elevation) dependent.
Content:

SUBROUTINE GRIDCK: Checks grid for correct structure.
SUBROUTINE GRIDW: Sets wavelength general grid.
SUBROUTINE GRIDZ: Sets altitude (elevation) general grid.

8.1.3.9  rad_tr.f

File rad_tr.f contains the code that performs the radiation transfer simulation once the environment conditions are set.

Content:

SUBROUTINE PS2STR: Computes 2 stream radiative transfer approximation.
SUBROUTINE TRIDAG: Solves tridiagonal matrix system.
SUBROUTINE RTLINK: Transforms data to the desired format, then call ps2str.

8.1.3.10  tuvutl.f

File tuvutl.f contains utility routines that compute some general quantity or perform some general manipulation on data.

Content:

SUBROUTINE ADDPNT: Adds a point in a pair of related ordered grids.
SUBROUTINE ADMPNT: Adds multiple points in a pair of related ordered grids.
FUNCTION FSUM: Sums values in an array.
SUBROUTINE INTER1: Interpolates data given on one grid on another grid.
SUBROUTINE INTER2: Idem
SUBROUTINE INTER3: Idem
SUBROUTINE SPHERS: Computes slant path over vertical depth in spherical geom.
SUBROUTINE SUNDIS: Computes Earth-Sun distance variation for a given date.
SUBROUTINE ZENITH: Computes solar zenith and azimuth for a given date/place.
SUBROUTINE ZERO1: Initializes a 1-dim array with zeros.
SUBROUTINE ZERO2: Initializes a 2-dim array with zeros.
FUNCTION LOCREA: Locates where a real values stands in an ordered list.
SUBROUTINE JINTER: Computes J-value by interpolation from previously computed values
SUBROUTINE INTPOL: Performs a 2nd-order polynomial 3-points interpolation.
SUBROUTINE GTNGRP: Reads a group of lines giving all values for a given a variable.
SUBROUTINE GTNLIN: Reads a group of values on a line.
SUBROUTINE SPLLIN: Splits a line in elements.
8.1.3.11 outpaw.f

File outpaw.f contains code to save results in a database format defined by the CERN (European Organization for Particle Physics) computer libraries. The format is the RZ HBOOK n-tuple format. (See http://wwwinfo.cern.ch/asdoc/).

Content:

   SUBROUTINE FORPAW: Saves results on an HBOOK-type database (CERN libraries).
   SUBROUTINE TRADAT: Aggregates all results in a single COMMON for HBOOK.

8.1.3.12 tuvold.f

File tuvold.f contains code that defines photolysis reaction weighting functions in the original TUV code by Sasha Madronich. It is not used in this version.
Literature cited


