One Dimensional Coupled Model
User's Manual

Prepared By:
Arastoo P. Biazar

Atmospheric Science Program
University of Alabama in Huntsville
Huntsville, Alabama

January 13, 1994

Section I:
PROGRAM DESCRIPTION

What follows is the description of the main program routine, the subroutines that it calls and their main function, and model requirements in general.

Main routine:

The main routine is called "UAH1DMDL" and it consists of calls to 16 different subroutines. The main function of the first 15 subroutines is to provide the necessary input parameters and to do preparation for the model run. The 16th. subroutine called "UAHMDL" is the main body of the model and will do the required simulation. Subroutine names and arguments are:

SETUP .....opens the i/o files and library file
STAMP(ISTAMP).....prints out the current date and time to the unit ISTAMP
RDINPT(XTIME).....reads title and gear parameters
RDSPEC .....reads optional list of species to define order in printout
RDREAC .....reads reaction specifications, optional rate constants
RDRKA.....reads reaction rate constants (from the input file)
RDSOLAR.....reads photolysis rate constants
RDCCON.....reads species whose concentrations are constant
RDNDIS.....not used, (read species not subject to dispersion)
RDDISP .....not used, (read species subject to dispersion)
REORDR .....reorders species list
RDNAMES.....reads in the name of the files - met and flux data update files
RDFLUXA.....reads species with flux at the boundary
RDFLUXB.....reads species with deposition velocity at the boundary
RDFLUXC.....reads species with flux contribution to all levels
UAHMDL(XTIME).....solves equations by gear method

A description of each of these subroutines will follow.

Subroutine SETUP:

This subroutine is the only interactive part of the program. The inputs can either be typed in at the terminal, if the program is running interactively, or can be furnished in a file assigned to the standard input unit (i.e., unit 5). This routine requires the answers to three questions regarding debug option, input/output file names, and the reaction library file name. The three lines of input that satisfies this routine are:

Line 1: The user is prompted for the DEBUG option. The answer can be either "y" for "yes" or "n" for "no". If the answer is "yes", then the debug option will be turned on and the model output will be dumped to the screen. If

the answer is "no", then the debug option will be turned off and the output delivered to the output file.

Line 2: The user is prompted for the input/output filename. The name that user supplies here (with no extension) will be used for both input and output files with different extensions. Three files will be opened:

File 1: an input file with an extension of "IN".
File 2: an output file with an extension of "OUT".
File 3: an output file with an extension of "EQT".

For example if user supplies the name "TEST", then a file named "TEST.IN" will be opened as input file and must contain all the required input parameters. A file named "TEST.OUT" will be opened to deliver the output information. This file will contain specific information about the simulation. The third file to be opened is called "TEST.EQT" which is an output file and will contain the concentration values at equal time periods.

Line 3: The user is prompted for the reaction library filename. This name should be the complete filename. This file is an input file and must be supplied by the user. This file contains some of the Gear parameters plus a complete list of the reactions and their type and reaction rate constants. We will refer to this file as the permanent library file.

Subroutine RDINPT(XTIME):

This subroutine reads in the title of the run and other input parameters from both input files (i.e., parameter input file and permanent library file). Description of each will follow in the section describing the input/output files.

Subroutine RDSPEC:

This subroutine reads optional list of species from the parameter input file to define order in printout. Species names must be in exact form and spelling as they appear in the permanent library file. The names are separated by a space between them. Each line can contain as many names as the user wishes to write.

Subroutine RDREAC:

This subroutine reads reaction specifications. The subroutine first reads the reaction labels from parameter input file. These are the reactions that user wishes to include in the simulation. Then the permanent reaction library is searched for the reaction lists with these labels. If they are found they will be included in the simulation otherwise an error message will be displayed. By inclusion we mean that the subroutine RDREAC reads the complete list of the reaction. This include the reactants and the products of the reaction and rate constants or the parameters necessary to calculate the rate constants. Subroutine CALCR calculates the rate constants that need to be calculated every update time.

Subroutine RDRKA:

This subroutine reads reaction rate constants (from the parameter input file). It reads reaction labels followed by a reaction rate constant. This option allows the user specify reaction rate constants different from that of the permanent library.

Subroutine RDSOLAR:

This subroutine reads the time profile of the photolysis rates from the file called SOLARTC.IN. Photolysis rate constants are read only once at the beginning of the simulation and stored in an array called SRATE. Then each update interval subroutine UPDATE6 updates the rate constants by interpolation.

Subroutine RDCCON:

This subroutine reads species whose concentrations are constant during the course of the simulation. This means that a differential equation regarding the chemical reaction of these species will not be set but they will be treated for vertical diffusion and transport. They also will be updated if necessary.

Subroutine RDNDIS:

This subroutine is here for historical reason. It may be used later.

Subroutine RDDIS:

This subroutine is here for historical reason. It may be used later.

Subroutine REORDR:

This subroutine reorders the species array so that the species with the constant concentration will be moved to the end of the array. This is useful since it adds to the efficiency of the model when solving for stiff differential equations.

Subroutine RDNAMES:

This subroutine reads the name of the files containing the time series of meteorological data and boundary and incoming flux data. The first file contains the meteorological data. The second file contains the boundary fluxes, species deposition velocities, and flux of the pollutants at different vertical levels.

Subroutine RDFLUXA:

This subroutine reads species with a flux at the boundary from the parameter input file. The name and order of the species here must match the data in the chemistry update file. The flux unit right now is "(#/CM**3).CM/S" which can easily be changed to "PPM.CM/S" by a little modification to the code.

Subroutine RDFLUXB:

This subroutine reads species with a deposition velocity at the boundary from the parameter input file. The name and the order of the species should match that of the chemistry update file. The unit is cm/s.

Subroutine RDFLUXC:

This subroutine reads the name of the species with an incoming flux between two vertical levels in the model. The name and the order of the species should match that of the chemistry update file. The two vertical levels are also included in the update file.

Subroutine UAHMDL:

This subroutine is the main body of the model. It integrate the model equations and will terminate after the time period specified by the user.

Section II:
DESCRIPTION OF THE INPUT AND OUTPUT FILES

There are eight input files used by this model and three output files created by the model. These files are:

(A) Input files:

1) Interactive input file
2) Control parameter input file
3) Permanent library file
4) Meteorological data file
5) Chemistry data file
6) Photolysis rate constants file -- solar file
7) Initialization file -- initial concentration data file
8) Restart binary data file

(B) Output files:

1) Information output file
2) Concentration output file
3) Restart binary data file -- End of the run concentration file

1) Interactive input file:

This an optional file. The information requested by the model interactively can be provided in several ways. One way to furnish the required information is to write them into a file and then assign this file to be the standard input unit. The format of this file is:

Line 1: Either "y" or "n" for the answer to debug option
Line 2: Name of the control parameter input line
Line 3: Name of the permanent library file

2) Control parameter input file

The name of this file is given to the program interactively. This file contains the definition of the input parameters for 1d-coupled model. The contents of the file are as follows:

LINE #1:

VARIABLE NAME:	TITLE
VARIABLE TYPE:	REAL*8 TITLE(10) COMMON/GEAR/TITLE
USAGE:	An array of 10 containing the title for the run

LINE #2:

VARIABLE NAME: TMAX
VARIABLE TYPE: REAL*8 TMAX
COMMON/GEAR/TMAX
USAGE: Simulation time in hours. After read statement it is converted to seconds. TMAX = TMAX*3600.

LINE #3:

VARIABLE NAME: XEQT
VARIABLE TYPE: REAL XEQT
LOCAL TO SUBROUTINE RDINPT
USAGE: Number of species to be printed at equal time intervals in ".EQT" file.
NEQT = INTEGER(XEQT)
COMMON/GEAR/NEQT

LINE #4:

VARIABLE NAME: IRESTART
VARIABLE TYPE: INTEGER IRESTART
COMMON/RDINPT/IRESTART
USAGE: Indicates how the model to be initialized.

IRESTART = 0	->	Not a restart
IRESTART = 1	->	Restart with one file
IRESTART = 2	->	Restart with two files

If IRESTART=0 then model reads the initial values from a file supplied by the user.

If IRESTART=1 then model reads the initial values from a file called 'TCRESTART.BIN' which is a binary file created by the model at the end of each run.

If IRESTART=2 then model first reads the initial values from 'TCRESTART.BIN' file and then reads other values from a file supplied by the user. Content of this file can be a modification of the concentration values or some other new values not presented in 'TCRESTART.BIN' file.

LINE #5:

VARIABLE NAME: RUNTYPE
VARIABLE TYPE: INTEGER RUNTYPE
COMMON/UAHGEAR/RUNTYPE
USAGE: Indicates the type of experiment.

RUNTYPE = 1	->	DISPERSION+CHEM. REAC.
RUNTYPE = 2	->	CHEM. REAC. ONLY
RUNTYPE = 3	->	DISPERSION ONLY

If RUNTYPE=1 then both chemical reactions and dispersion are being considered and the full model is implemented.

If RUNTYPE=2 only chemical reactions are being considered and the dispersion mechanism is turned off.

If RUNTYPE=3 only dispersion is being considered and the chemical reactions mechanism is turned off.

LINE #6:

VARIABLE NAME: TVFLAG
VARIABLE TYPE: LOGICAL TVFLAG
USAGE: TVFLAG determines if temperature varies during the run or it is a constant.

If TVFLAG=.TRUE. then temperature profile is being updated every update time (TUPDATE time interval).

If TVFLAG=.FALSE. then temperature profile is set to a constant value and the next line in the input file must be the constant temperature value.

LINE #7:

VARIABLE NAME: PVFLAG
VARIABLE TYPE: LOGICAL PVFLAG
USAGE: PVFLAG determines if pressure varies during the run or it is a constant.

If PVFLAG=.TRUE. then pressure profile is being updated every update time (TUPDATE time interval).

If PVFLAG=.FALSE. then pressure profile is set to a constant value and the next line in the input file must be the constant pressure value.

LINE #8:

VARIABLE NAME: IUPDATE
VARIABLE TYPE: INTEGER IUPDATE
USAGE: IUPDATE is the update indicator, it determines the value of logical name NOUPDATE

If IUPDATE=0 then NOUPDATE=.TRUE. and the subroutine UPDATE is bypassed.

If IUPDATE=1 then NOUPDATE=.FALSE. and the subroutine UPDATE is being called every TUPDATE time interval.

LINE #9:

VARIABLE NAME: TUPDATE
VARIABLE TYPE: REAL*8 TUPDATE
USAGE: Time interval for updating the variable data; such as pressure, temperature and boundary fluxes.

LINE #10...to the END-OF-FILE:

From here to the end of the file inputs are in a block format. Each block starts with a comment line followed by as many line of input as necessary. The block ends with the symbol "*END" in a separate line by itself indicating the end of the data. A block with only the comment line followed by a line with "*END" means that the block does not contain any data. Blocks are as follows:

BLOCK #1:

This block contains the name of the species to be printed. Species names must be in exact form and spelling as they appear in the permanent library file. The names are separated by a space between them. Each line can contain as many names as the user wishes to write. Subroutine "RDSPEC" reads this block.

BLOCK #2:

This block contains the reaction labels. Only the reactions with these labels will be included in the simulation. The program reads the permanent reaction library and includes the reactions with matching labels. The labels here must be in exact form as they appear in the permanent library. Subroutine "RDREAC" reads this block.

BLOCK #3:

This block contains the list of reaction labels followed by a reaction rate constant. This block is used if there are reactions with rate constants different from that of the permanent library. Subroutine "RDRKA" reads this block.

BLOCK #4:

This block contains the name of species which have constant concentration during the simulation. This means that a differential equation regarding the chemical reaction of these species will not be set. But they will be treated for vertical diffusion and transport. They also will be updated if necessary. Subroutine "RDCCON" reads this block.

BLOCK #5:

This block is here for historical reasons. It may be used later. Subroutine "RDNDIS" reads this block.

BLOCK #6:

This block is here for historical reasons. It may be used later. Subroutine "RDDIS" reads this block.

BLOCK #7:

This block contains the name of the files containing the time series of meteorological data and boundary and incoming flux data. The first line of this block starts with integer number "1" followed by a space and the name of the file containing the meteorological data. The second line starts with integer number "2" followed by a space and the name of the file containing the boundary fluxes, species deposition velocities, and flux of the pollutants at different vertical levels. Subroutine "RDNAMES" reads this block.

BLOCK # 8:

This block contains the name of the species with boundary flux. The name and order of the species here must match the data in the chemistry update file. The flux unit right now is "(#/CM**3).CM/S" which can easily be changed to "PPM.CM/S" by a little modification to the code. Subroutine "RDFLUXA" reads this block.

BLOCK #9:

This block contains the name of the species with deposition velocity at the boundary. The name and the order of the species should match that of the chemistry update file. The unit is CM/S. Subroutine "RDFLUXB" reads this block.

BLOCK #10:

This block contains the name of the species with an incoming flux between two vertical levels in the model. The name and the order of the species should match that of the chemistry update file. The two vertical levels are also included in the update file.

BLOCK #11:

This block contains the name of the file containing the species initial values for all the grids in the model.

3) Permanent Reaction library input file

The name of this file is given to the program interactively. The permanent reaction library input file contains some of the control parameters for the Gear solver and a complete list of the reactions and factors used in calculations of reaction rate constants. The following is the line by line description of the contents of this file.

Line 1:

VARIABLE NAME:	EPS	
VARIABLE TYPE:	REAL*8	EPS

USAGE: Tolerance for convergence in the Gear routine. This tolerance can be changed internally in Gear if the convergence is not achieved within the specified iteration limit.

Line 2:

VARIABLE NAME: HMIN
VARIABLE TYPE: REAL*8 HMIN
USAGE: Minimum time step allowed in the integration. Time step in the Gear routine cannot be reduced to a value less than HMIN. If this value is reached, the program will terminate with an error message.

Line 3:

VARIABLE NAME: HMAX
VARIABLE TYPE: REAL*8 HMAX
USAGE: Maximum integration time step to be taken in the simulation.

Line 4:

VARIABLE NAME: NOUT
VARIABLE TYPE: INTEGER NOUT
USAGE: Number of species in the output. The concentration of the first NOUT members of the species list will be printed in the "*.DAT" file.

Line 5:

VARIABLE NAME: NTEST
VARIABLE TYPE: INTEGER NTEST
USAGE: For historical reasons.

Line 6:

VARIABLE NAME: NWRPLO
VARIABLE TYPE: INTEGER NWRPLO
USAGE: For historical reasons.

Line 7:

VARIABLE NAME: MF
VARIABLE TYPE: INTEGER MF
USAGE: Method indicator for Gear routine. This variable indicates what method to be used in Gear routine. Notice that we use a variable order multi-step method. This means the order of the method (number of steps in the multi-step method) changes internally to achieve the required accuracy.

MF	**METHOD USED**

0 An ADAMS Predictor-Corrector is used. ADAMS method is a multistep method. It can be both implicit and explicit. If the implicit method is used to improve the approximation obtained by an explicit method, the combination is called "Predictor-Corrector" method. Thus with choice of MF=0, the ADAMS explicit method is used to predict the approximation, then ADAMS implicit method is used to correct it.

1 A multi-step method suitable for stiff system is used. It will also work for non-stiff systems. However the user must provide a subroutine "PEDERV" which evaluates the partial derivatives of the differential equations with respect to the YC's. This is done by call PEDERV(XTIME,YC,PW,M). Where PW is an N by N working array which must be set to the partial of the i-th equation with respect to the j-th dependent variable in PW(i,j). PW is actually an M by M array, where M is the value of N used on the first call to DIFSUB.

2 The same as case I, except that this subroutine computes the partial derivatives by numerical differencing of the derivatives. Hence PEDERV is not called.

Line 8:

VARIABLE NAME: NLARGE
VARIABLE TYPE: INTEGER NLARGE
USAGE: This parameter indicates the method of matrix inversion for the matrix of partial derivatives.
If NLARGE=0 then the inverse of matrix is found directly (if non-singular).
If NLARGE is not zero then the program performs L/U decomposition on matrix of partial derivatives.

Line 9:

VARIABLE NAME: TINC
VARIABLE TYPE: REAL*8 TINC
USAGE: This parameter indicates the TINC is the output time interval (in hrs).

Line 10 through the end of the file:

Line 10 contains the comment line. Following this comment line, each line contains a reaction label followed by the reaction list, type of the reaction, and necessary parameters for calculating the rate constants. There are several states in reading the reaction list. The first state is when the reaction label is being read. It is checked against the list of reaction labels and an error message is printed if a match is not found, i.e., it is an undefined reaction label.

The second state is reading reactants. Reactants are read until an '=' is encountered. '+'s, which separate reactants, are optional and ignored.
The third state is reading products. '+'s are ignored.
The products are followed by a ';', which indicates the start of the fourth state. In the fourth state the program reads the type of rate constant and from that determines how many more variables to read for that type of rate constant calculation. It then reads that many variables and puts them into the array RKMOD. Then the program reads the order of the reaction (i.e., 1 for first order reactions, 2 for 2nd. order reactions and so forth).
The fifth state occurs when a '000'O occurs, which means the end of the data line has been reached - the next line is read.

Rate constants for each type is calculated in the following way:

CASE #1

Reactions that are dependent on solar intensity. Rate constant calculated elsewhere and are inputted from SOLARTC.IN file.

CASE #2

Reactions which have constant rate constants. Read 1 subsequent number; that is the rate constant.

CASE #3

Reactions which are temperature dependent. Need 2 subsequent numbers; 1st one is A (pre-exponential factor) 2nd one is E (activation energy).
The calculation is according to the ARRHENIUS EQUATION:

K = A*exp(-E/RT)

CASE #4

Pressure-dependent reactions (set of 5 with common equation).
Need 4 subsequent numbers:

1ST = KO(300), 2ND = N, 3RD = KI(300), 4TH = M

KA = KO^{300}

KB = N

KC = KI^{300}

KD = M

$$KO(T) = KO^{300} * \left(\frac{T}{300}\right)^{-N}$$

$$KI(T) = KI^{300} * \left(\frac{T}{300}\right)^{-M}$$

$$K = \left[\frac{KO(T) * PRES}{1 + \left(\frac{KO(T) * PRES}{KI(T)} \right)} \right] * 0.6^{\left\{ 1 + \left[Log_{10} \left(\frac{KO(T) * PRES}{KI(T)} \right) \right]^2 \right\}^{-1}}$$

CASE #5

These are unimolecular reactions.
Read 3 subsequent numbers:
1ST = number of reaction that this is the reverse of
2ND = A, 3RD = B

$$K = KR * A * e^{\left(\frac{-B}{T} \right)}$$

where

$$KR = \left[\frac{KO(T) * PRES}{1 + \left(\frac{KO(T) * PRES}{KI(T)} \right)} \right] * 0.6^{\left\{ 1 + \left[Log_{10} \left(\frac{KO(T) * PRES}{KI(T)} \right) \right]^2 \right\}^{-1}}$$

and

$$KO(T) = AA * \left(\frac{T}{300} \right)^{-BB}$$

$$KI(T) = CC * \left(\frac{T}{300} \right)^{-DD}$$

The values of AA, BB, CC and DD depend on the value of the first input parameter. We let the first parameter to be named NREV. Then:

$$NREV = 20 \Rightarrow \begin{cases} AA = 1.8 \times 10^{-31} \\ BB = 3.2 \\ CC = 4.7 \times 10^{-12} \\ DD = 1.4 \end{cases}$$

and

$$NREV = 22 \Rightarrow \begin{cases} AA = 2.2 \times 10^{-30} \\ BB = 4.3 \\ CC = 1.5 \times 10^{-12} \\ DD = 0.5 \end{cases}$$

CASE #6

Special CASE #1.
Individual calculation for
O3P + O2 --> O3

$$K = [PRES] * 6.0 * 10^{-34} * \left(\frac{T}{300}\right)^{-2.3}$$

CASE #7

Special CASE #2
Individual calculation for
HO2 + HO2 --> H2O2

$$K = 2.2 * 10^{-13} * e^{\left(\frac{620}{T}\right)} + 1.9 * 10^{-33} * [PRES] * e^{\left(\frac{980}{T}\right)}$$

CASE #8

Special CASE #3
Individual calculation for
HO2 + HO2 + H2O --> H2O2

$$K = 3.08 * 10^{-34} * e^{\left(\frac{2820}{T}\right)} + 2.66 * 10^{-54} * [PRES] * e^{\left(\frac{3180}{T}\right)}$$

CASE #9

Special CASE #4
Individual calculation for
HO + HNO3 -> NO3 + H2O

KO=7.2E-15*EXP(785./T)
K2=4.1E-16*EXP(1440./T)
K3=1.9E-33*EXP(725./T)*PRES
K = KO + K3/(1+ K3/K2)

CASE #10

Special CASE #5
Individual calculation for
CO + HO -> HO2 + CO2

K=1.5E-13 * (1. + 2.439E-20 * PRES)

CASE #11

Special CASE #6
Individual calculation for
HO + HO2 -->

$$K = 1.7 * 10^{-11} * e^{\left(\frac{416}{TA}\right)} + 3.0 * 10^{-31} * [PRES] * e^{\left(\frac{500}{TA}\right)}$$

4) Meteorological data file

This file contains the necessary meteorological data. This file is created by running the Mesoscale Boundary Layer in 1_D mode. The contents of this file are as follows:

Line 1:

Number of vertical grid points KZ, followed by the delta-t (time increment used in mesoscale model). These values are read only once by subroutine INITDIF.

Line 2...through line 2+KZ:

The height of the vertical levels. The heights are in centimeter (CM).

Line KZ+2+1:

Time of the output. This is the mesoscale model simulation time in minutes.

Line KZ+2+2:

USTAR, PHISTAR, and ROOF.
USTAR is the friction velocity in CM/S.
PHISTAR another friction quantity which is not being used at the present time.
ROOF is the planetary boundary layer height in CM.

Line KZ+2+3 ... through KZ+2+3+KZ:

Each of the following lines contains THETAP, PIP, QVP, and ALPHA for each vertical grid point.
THETAP is the potential temperature in degrees Kelvin, it will be converted to temperature by the function THETATOT.
Since potential temperature is calculated according to Poisson's equation:

THETA = T * (P0/P)**(R/CP)

thus T = THETA * (P/P0)**(R/CP)

Units of input must be as follows:

THETA: IN DEGREES K
P: IN MILIBARS

Units of output is as follows:

THETATOT: In degrees Kelvin

Constants used:

P0 = 1000. MILIBARS
CP = 1.005E+03 J/(KG K)
= 1.005E+07 CM**2/(S**2 K)
RD = 287 J/(KG K)
RD/CP = .286

Therefore the equation for the conversion will be:

T = THETA * (P/1000.)**0.286

PIP is scaled pressure (PI) use the EXNER function to get the unscaled pressure in MILIBARS:

P = P0 * (PIP/(CP*10000.))**(CP/RD)

WHERE

CP = 1004. M**2/(S**2 . K)

CP/RD = 3.4965

P0 = 1013.25 MB

and 10000. is the unit correction factor for M**2 to CM**2

QVP is specific humidity (MVAPOR/MMOIST-AIR) IN G/G

ALPHA is the diffusion coefficient in CM**2/S

The last KZ+2 line of this file will be repeated. When it is time to update the meteorological data this file is being read again to update the values.

5) Chemistry data file

This file contains the information about the chemical species and their fluxes. The file can be created with the help of a preprocessor. The preprocessor program is called "CHEMBLD.for". The chemistry data file contains 1) the flux of species at the boundary, 2) the deposition velocities, and 3) flux of the species between levels 1 and 2 of the model. Each of these three elements can be and must omitted according to the specification in the parameter-input file. For example if there is no species with incoming flux between two model levels, the control parameter data file does not have any entry in the corresponding section. Thus, chemistry data file also should not contain any entry. In another word, control parameter data file contains the list of the species whose values are read from chemistry data file. For the same reason the order of the values in the chemistry data file must match the order of the species list in the control parameter data file. The elements of the chemistry data file are as follows:

1) A. A HEADER LINE FOR SPECIES WITH FLUXES AT THE BOUNDARY
   B. VALUE OF THE FLUX IN ((#/CM**3).(CM/S))

2) A. A HEADER LINE FOR DEPOSITION VELOCITIES
   B. VALUES IN (CM/S)

3) A. A HEADER LINE FOR SPECIES WITH FLUX TO THE LEVELS 1 TO 2
   B. VALUES IN ((#/CM**3)/S)

The number of entries (lines) for each block depends on the number of species specified in the control parameter data file.

6) Photolysis rate constants file -- solar file

This file contains the time profile of the photolysis rate constants. At the present time the file is called "SOLARTC.IN" and can be created by running the program "SOLAR.FOR". Program SOLAR calculates the photolysis rate constants for different species according to the geographical area and time of the day by calculating the solar radiation intensity for a given point in space and time and using a table of quantum yields for species.

The content of the file SOLARTC.IN is:

Line 1: Comment line

Line 2: Starting time, Ending time, Time increment.

Starting local time is in 2400 format. Ending time is also in 2400 format except that this is the accumulated time, thus for a four day simulation (96 Hrs) the Starting time is 600 and the Ending time is 10200. Time increment is in minutes.

Line 3...

From line 3 to the end of the file the same format for each of the species will follow.

The header line contains the reaction label and the name of the species in the same form as in the permanent reaction library.

The header line is followed by several lines of data.

Each of these lines have three elements, the Time, solar zenith angle, and the photochemical reaction first order rate constant for the species in the header line at the time indicated. The unit for the rate constant is 1/Sec.

7) Initialization file -- initial concentration data file

This file contains the initial values for species concentrations. This file is an ASCII file and be created with a text editor. The first line of this file is a comment line. Each of the following blocks have the name of the species in a separate line followed by the species initial concentration for each grid point (starting with grid point 1 at the lowest model level). This file sometimes is used in conjunction with TCRESTART.BIN file which is a binary file created by the model at the end of each run.

8) Restart binary data file

We can use the restart binary file to restart the model from the point that model was last terminated. We can also change some of the concentration values when restarting the model. Model automatically creates this binary file at the end of each simulation. This binary file that at the present time is called "TCRESTART.BIN" is optional and it is used if IRESTART is 1 or 2.