PLUVIUS

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

The views, opinions and conclusions contained in this report are those of the contractor and do not necessarily represent those of the United States Government or the United States Department of Energy.

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
Under Contract DE-AC06-76RLO 1830

Printed in the United States of America
Available from
National Technical Information Service
United States Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151

Price: Printed Copy \$_____*: Microfiche \$3.00

*Pages	NTIS
	Selling Price
001-025	\$4.00
026-050	\$4.50
051-075	\$5.25
076-100	\$6.00
101-125	\$6.50
126-150	\$7.25
151-175	\$8.00
176-200	\$9.00
201-225	\$9.25
226-250	\$9.50
251-275	\$10.75
276-300	\$11.00

Addendum

Shortly after publishing this manual we discovered that I had made a significant error in my parameterization of rainfall velocity in Example Application 2. This error also affected the virtual diffusivity parameterization. To correct for this we suggest replacing w in Table 20 with

$$w = -4116 [\text{rain}]^{.125} ,$$

Also, equation (33) should be replaced by

$$D_{\text{rain}} = 100,000 + 1.116 \times 10^6 \Delta t [\text{rain}]^{.25} .$$

Our experience has shown that these changes have the following results:

- The showery nature shown in Figure 6 is dampened appreciably.
- The higher fall velocity tends to make the numerical solution less stable and more time consuming.

We suggest the following procedures to reduce the second effects:

- Eliminate the portion of CONC which provides for the ultimate evaporation of rain. This will allow minute amounts of "rain" to diffuse below the rainshaft, but will eliminate the step function below the virga.
- Put in an alternative evaporation term in GEN for low rain situations to prevent numerical overshoot arising from this term. An example:

```
C    RATE OF GAIN OF VAPOR AND CLOUD WATER
      DUMMY = D2* C(I,3)**.65
      IF (C(I,3).LT.1.E-12) DUMMY = C(I,3)/300
      G(I,2) = D1*C(I,10) + DUMMY
```

The above fixes are not intended to be permanent or long-standing. They are meant primarily to give the user a working code from which he can readily incorporate his own parameterizations in the most rapid and expedient manner.

From the above it should be evident that PLUVIUS is not yet a totally docile system; we are currently developing a MOD 4.0 version, which should be somewhat more satisfactory in this respect as well as being more accurate and versatile for scientific analysis.

- PLUVIUS -
A GENERALIZED ONE-DIMENSIONAL MODEL OF
REACTIVE POLLUTANT BEHAVIOR, INCLUDING DRY
DEPOSITION, PRECIPITATION FORMATION, AND WET REMOVAL

J. M. Hales

September 1981

Prepared for
the U.S. Environmental Protection Agency
under a Related Services Agreement
with the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Richland, Washington 99352

Foreword

The computer code described in this report constitutes the most recent in a series of developing models, the first of which was created at Stockholm University during 1979. There has been a continuous succession of improvements since that time; and the current version, titled PLUVIUS, MOD 3.1, represents a working code whose capabilities are significantly stronger than its predecessors. Although future improvements can be expected to occur, MOD 3.1 is in a form where it is appropriate for dissemination to the scientific community at large. The present report has been prepared to fulfill this end.

Most of us find the proposition of applying someone else's computer code at least mildly distasteful, and the present manual and code have been prepared so as to reduce this effect as much as possible. The program is subdivided into basic modules, whose coordination within the code is presented in flow-chart form. Input and output variables are given in consistent units, and the program listings are documented at critical points. In addition, two example code executions are described, along with corresponding input and representative output. As a first step in code application the user should execute these examples on his own machine. Progressive modifications from these base cases will lead to the creation of working versions for user application in the most rapid and expedient manner.

Although considerable working experience with PLUVIUS has been gained at this laboratory, the high versatility of the code has precluded the comprehensive testing of all combinations of features. It is possible that some as-of-yet undetected snag exists in the system. Because of this I would deeply appreciate receiving any reports of experiences with the code which may arise from the user community.

The scientific ideas embodied in the PLUVIUS code are by no means solely mine, and I would like to express my sincere thanks in particular to the following individuals for their helpful discussions in this regard:

Mr. Richard Easter	Pacific Northwest Laboratory
Dr. Lennart Granat	Stockholm University
Dr. Henning Rodhe	Stockholm University
Dr. Bryan Scott	Pacific Northwest Laboratory
Dr. Hilding Sundqvist	Stockholm University

This work has been supported financially by Stockholm University and under the MAP3S/RAINE program by the U.S. Environmental Protection Agency and the U.S. Department of Energy. Three people in particular, Dr. Granat, Dr. David Ballantine and Mr. Robert Beadle, have acted to make this support possible, and I express my sincere gratitude to them for their active efforts in this area.

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	FORMULATION OF GOVERNING EQUATIONS AND BOUNDARY CONDITIONS . .	3
	A. Governing Equations Describing Material-Balance Relationships	3
	B. Initial and Boundary Conditions for Material-Balance Relationships	5
	C. Governing Equation Describing Energy-Balance Relationship .	7
	D. Initial and Boundary Conditions for Energy-Balance Relationship	10
III.	CODE CHARACTERISTICS -- AN OVERVIEW	11
	A. Initial Rules and Conventions	11
	B. Data Input	13
	C. Modular Construction	16
	1. Grid Structure Control	17
	2. Initial and Boundary Conditions	22
	3. Transport Properties	26
	4. Physical and Chemical Reaction	30
	5. Thermodynamic Properties	30
IV.	EXAMPLE APPLICATION 1: SIMPLE AIR CHEMISTRY AND DRY DEPOSITION	43
V.	EXAMPLE APPLICATION 2: REACTIVE SCAVENGING IN AN OROGRAPHIC STORM	53
	A. General Description	53
	B. Ordering of Variables	56
	C. Grid Structure and Dynamic Forcing	59
	D. Initial and Boundary Conditions	63
	E. Transport Properties	63
	F. Physical and Chemical Reaction	67
	G. Thermodynamic Properties	67
	H. Class III Subroutines	71
	I. Deck Structure and Input Data for Example Application 2 .	76
	J. Example Output for Example Application 2	78

VI. PROBLEMS WITH PLUVIUS: SOME PRACTICAL DIFFICULTIES AND SUGGESTIONS FOR THEIR RESOLUTION	95
REFERENCES	98
APPENDIX A	
APPENDIX B	

LIST OF FIGURES

<u>Number</u>		<u>Page</u>
1	Schematic of Computation Grid	19
2	Results of Example 1 Computations at 3000 Seconds of Simulation Time From Initial State	51
3	Interconversion Diagram for Water	54
4	Interconversion Diagram for Sulfur	55
5	Outlines of Cloud and Precipitation Regions for Model Storm	60
6	Computed Precipitation Rates for Model Storm	89
7	Computed Primary (solid) and Secondary (dashed) Sulfate Concentrations in Ground-Level Rainwater; 0.1-micron Aerosol Case	90
8	Computed Primary (solid) and Secondary (dashed) Sulfate Concentrations in Ground-Level Rainwater; 0.3-micron Aerosol Case	91
B.1	Flowchart for Main Program PLUVIUS	B-13

LIST OF TABLES

<u>Number</u>		<u>Page</u>
1	Significances of Bypass Control Variables	15
2	Summary of Species Properties for Simple Example Code Application	18
3	Subroutine GRDSET: Typical Listing	20
4	Subroutine TIMESP: Typical Listing	23
5	Subroutine VERT: Typical Listing	24
6	Subroutine BOUND: Typical Listing	25
7	Subroutine CONC: Typical Listing	27
8	Subroutine DIFF: Typical Listing	28
9	Subroutine SETTLE: Typical Listing	29
10	Subroutine GEN: Typical Listing	31
11	Subroutine TEMPF: Typical Listing	33
12	Subroutine PRESSF: Typical Listing	34
13	Class II Subroutines and Their Functions	35
14	Summary of Code Nomenclature	36
15	Input Data for Example 1	44
16	Computed Results for 0 Seconds of Model Time	46
17	Constrained Results for 0 Seconds of Model Time	47
18	Computed Results for 3000 Seconds of Model Time	48
19	Constrained Results for 3000 Seconds of Model Time	49
20	Summary of Parameterizations Used for Example Computation	57
21	Ordering and Initial Conditions for Example Application 2	58
22	Subroutine VERT: Listing for Example 2	61

<u>Number</u>		<u>Page</u>
23	Subroutine TIMESP: Listing for Example 2	62
24	Subroutine BOUND: Listing for Example 2	64
25	Subroutine CONC: Listing for Example 2	65
26	Subroutine DIFF: Listing for Example 2	68
27	Subroutine SETTLE: Listing for Example 2	69
28	Subroutine GEN: Listing for Example 2	70
29	Subroutine TEMPF: Listing for Example 2	72
30	Listings of the Class III Functions of SAT and DSAT	75
31	Listing of the Class III Subroutine HENRY	77
32	Input Data for Example 2	79
33	Initial Computed Variables for Example Application 2 0.1 Micron Aerosol Particles	80
34	Constrained Variables for Example Application 2 0.1 Micron Aerosol Particles	81
35	Initial Atmospheric Pressures and Air Concentrations	82
36	Computed Variables for Example Application 2 0.1 Micron Aerosol Particles	83
37	Constrained Variables for Example Application 2 0.1 Micron Aerosol Particles	84
38	Atmospheric Pressures and Air Concentrations at .1177E+05 Simulation Time	85
39	Computed Variables for Example Application 2 0.3 Micron Aerosol Particles	86
40	Constrained Variables for Example Application 2 0.3 Micron Aerosol Particles	87
B.1	PLUVIUS Program Listing	B-1
B.2	Listing of Class II Subroutine PRINT	B-8
B.3	Listing of Class II Subroutine TRIDAG	B-10

<u>Number</u>		<u>Page</u>
B.4	Listing of Class II Subroutine DERIV	B-11
B.5	Listing of Class II Subroutine SETVEC	B-12

- PLUVIUS -

A GENERALIZED ONE-DIMENSIONAL MODEL OF
REACTIVE POLLUTANT BEHAVIOR, INCLUDING DRY
DEPOSITION, PRECIPITATION FORMATION, AND WET REMOVAL

-Man is not capable of thought in any high degree, and even the most spiritual and highly cultivated of men habitually sees the world and himself through the lenses of delusive formulas and artless simplifications.

Hermann Hesse, *Steppenwolf*

I. INTRODUCTION

During recent years numerous investigations of atmospheric behavior have been performed using one-dimensional simplifications of the continuity equations for mass, energy, and momentum. Such methods necessarily sacrifice some aspects of reality in the context of our three-dimensional atmosphere; but quite often they lead to valuable insights which could not be obtained otherwise within the constraints of computational capabilities and human perception. In the context of the Steppenwolf's rather pessimistic observation, one-dimensional models (and indeed all other types of atmospheric interpretive models) create a type of lens through which atmospheric processes can be observed. They are imperfect lenses; but frequently they are capable of creating insights and concepts which would be largely impossible in their absence.

This statement is certainly a valid one with reference to air-pollutant transformation and removal processes. The chemistry and physics of these phenomena tend to be both poorly understood and mathematically complex, and the computational simplicity of one-dimensional modeling is thus a highly attractive feature. One example of such an application is the analysis of Omstedt and Rodhe (1978), who performed computations for the three-component system H_2S - SO_2 - $\text{SO}_4^{=}$ using various assumptions with regard to chemical reaction, deposition and vertical transport.

Previous computer codes applied for these purposes tend to be rather specific to the immediate requirements at hand, and it seems desirable to

possess a generalized code which would provide a flexible framework for numerous types of calculations. Specifically, there is a need for a code which would allow for:

- variable diffusion in height and in time
- generalized boundary conditions at both the top and bottom of the computational grid, allowing a versatile characterization of the deposition/resuspension process
- flexible vertical grid spacing
- the capability to describe cloud and precipitation formation phenomena
- the versatile incorporation of aqueous-phase and gaseous-phase chemical conversion for multiple-component systems

and

- the capability to describe wet removal processes.

The present report describes a code which has been developed for this purpose, and the following chapters provide a user's manual for its practical application. This manual begins by formulating the one-dimensional conservation equations to be used by the code. It then gives a superficial description of the code, which is intended to facilitate its use for example calculations, which constitute the final portion of the main body of the report. More detailed description of the code, along with a discussion of its numerical procedures, is given in the Appendix.

II. FORMULATION OF GOVERNING EQUATIONS AND BOUNDARY CONDITIONS*

A. Governing Equations Describing Material-Balance Relationships

The starting point for this derivation is the conventional continuity equation for pollutant species k (cf. Bird, Stewart, and Lightfoot (1960)):

$$\frac{\partial c_k}{\partial t} = - \nabla \cdot (c_k \mathbf{v}_k) + G_k \quad (1)$$

$$k = 1, 2, 3, \dots, m$$

where c denotes molar concentration, $\mathbf{v}$ is the pollutant velocity vector and G is a volumetric generation term. Dropping the subscript k for convenience, applying conventional K-theory treatment, and expanding the divergence term yields the form

$$\frac{\partial c}{\partial t} = - \frac{\partial}{\partial x} (cu) - \frac{\partial}{\partial y} (cv) - \frac{\partial}{\partial z} (cw) + G ; \quad (2)$$

where

$$u = u^* - \frac{c^*}{c} D_x \frac{\partial}{\partial x} (c/c^*)$$

$$v = v^* - \frac{c^*}{c} D_y \frac{\partial}{\partial y} (c/c^*)$$

$$w = w^* - \frac{c^*}{c} D_z \frac{\partial}{\partial z} (c/c^*)$$

*The mathematics of this section are somewhat involved, and it is suggested that the reader skim this material initially and proceed rapidly to later sections describing code utilization. This section can be re-studied as necessary after one has had initial experience in executing the code.

Here c^* denotes the molar concentration of air and u^* , v^* and w^* pertain to the non-diffusive components of the individual pollutant velocities.

Let the corresponding components of air velocity be u_A , v_A , and w_A , and assume that

$$u = u_A \quad (3)$$

and

$$v = v_A$$

Also, let

$$w = w_A + w_r - \frac{c^*}{c} D_z \frac{\partial}{\partial z} (c/c^*) \quad , \quad (4)$$

where w_r is the nondiffusive component of the pollutant velocity relative to vertical air velocity; then

$$\begin{aligned} \frac{\partial c}{\partial t} = & - \frac{\partial}{\partial x} (cu_A) - \frac{\partial}{\partial y} (cv_A) - \frac{\partial}{\partial z} (cw_A) - \frac{\partial}{\partial z} (cw_r) + \\ & \frac{\partial}{\partial z} [c^* D_z \frac{\partial}{\partial z} (c/c^*)] + G \end{aligned} \quad (5)$$

If transients in density are small, one can write the equation of mass continuity for air as

$$\nabla \cdot (c^* \chi_A) = 0 \quad (6)$$

Incorporating this with (5) gives

$$\begin{aligned} \frac{\partial c}{\partial t} = & - u_A \frac{\partial c}{\partial x} - v_A \frac{\partial c}{\partial y} - w_A \frac{\partial c}{\partial z} - w_r \frac{\partial c}{\partial z} - c \frac{\partial w_r}{\partial z} \\ & + c \left\{ \frac{u_A}{c^*} \frac{\partial c^*}{\partial x} + \frac{v_A}{c^*} \frac{\partial c^*}{\partial y} + \frac{w_A}{c^*} \frac{\partial c^*}{\partial z} \right\} \end{aligned}$$

$$+ \frac{\partial}{\partial z} [c^* D_z \frac{\partial}{\partial z} (c/c^*)] + G \quad (7)$$

If one expresses the vertical air velocity as the sum of the change rate of terrain elevation w_{A_0} and some additional component w'_A , i.e.,

$$w_A = w_{A_0} + w'_A, \quad (8)$$

and defines in addition a "terrain-following" derivative

$$\frac{\hat{D}c}{\hat{D}t} = \frac{\partial c}{\partial t} + w_{A_0} \frac{\partial c}{\partial z} + u_A \frac{\partial c}{\partial x} + v_A \frac{\partial c}{\partial y} \quad (9)$$

then (7) becomes

$$\begin{aligned} \frac{\hat{D}c}{\hat{D}t} &= -(w_r + w'_A) \frac{\partial c}{\partial z} - c \frac{\partial w_r}{\partial z} + c \frac{w_A}{c^*} \frac{\partial c^*}{\partial z} + \frac{\partial}{\partial z} \left[c^* D_z \frac{\partial}{\partial z} (c/c^*) \right] + G \\ &= D_z \frac{\partial^2 c}{\partial z^2} + \left[\frac{\partial D_z}{\partial z} - \frac{D_z}{c^*} \frac{\partial c^*}{\partial z} - w_r - w'_A \right] \frac{\partial c}{\partial z} \\ &\quad + \left[\frac{D_z}{c^{*2}} \left(\frac{\partial c^*}{\partial z} \right)^2 - \frac{D_z}{c^*} \frac{\partial^2 c^*}{\partial z^2} + \frac{1}{c^*} (w_A - \frac{\partial D_z}{\partial z}) \frac{\partial c^*}{\partial z} - \frac{\partial w_r}{\partial z} \right] c + G \\ &= A1 \frac{\partial^2 c}{\partial z^2} + B1 \frac{\partial c}{\partial z} + C1 c + G. \end{aligned} \quad (10)$$

Here the total derivative $\hat{D}c/\hat{D}t$ can be visualized as the time rate of change of material in a vertical column of air moving with the velocity u_A , v_A , w_A . G is a catch-all term, providing for the combined effects of sources, reaction, and interphase transport.

B. Initial and Boundary Conditions For Material-Balance Relationships

Equation (10) is a second-order parabolic equation with two independent variables, and as such it requires two boundary conditions and one initial

condition for a complete formulation of the problem. The initial condition can be stated quite simply as

$$c(0, z) = \hat{c}(z) \quad (11)$$

where $\hat{c}$ is some specified initial concentration field.

Boundary conditions can be specified at the top and bottom of the concentration field, and can differ in form. Two forms chosen for the present application are:

- flux-specified conditions; that is, the flux of pollutant through the boundary is specified,

and

- concentration-specified conditions; that is, the concentration of pollutant at the boundary is specified.

Mathematically these conditions can be expressed as follows:

Upper boundary condition

- flux specified:

$$\hat{f}_T(t) = [w_r(t, h) + w'_A(t, h)] c(t, h) - D_z(t, h) c^* \frac{\partial c/c^*}{\partial z} \Big|_h \quad (12)$$

- concentration specified:

$$c(t, h) = \hat{c}_T(t) \quad (13)$$

Lower boundary condition

- flux specified:

$$\hat{f}_B(t) = w_r(t, 0) c(t, 0) - D_z(t, 0) c^* \frac{\partial c/c^*}{\partial z} \Big|_0 \quad (14)$$

- concentration specified:

$$c(t,0) = \hat{c}_B(t). \quad (15)$$

Here $\hat{c}_T$, $\hat{c}_B$, $\hat{f}_T$, and $\hat{f}_B$ pertain to specified pollutant concentrations and fluxes at the top and bottom of the computational domain. It is important to note that one set of initial and boundary conditions is required for every species to be computed.

C. Governing Equation Describing Energy-Balance Relationship

It is important to note at this point that energy-balance considerations are necessary to the current model only when one desires to compute temperature distributions within the atmosphere. In view of this the following formulation is unnecessary for many applications; and indeed the model's computer code allows the associated operations to be bypassed when appropriate.

The starting point for formulation of the appropriate energy-balance equation can be obtained from equation (10.1-19) from Bird, Stewart, and Lightfoot (1960). Expressing this equation on a molar basis, neglecting viscous generation, and adding a latent-heat term one obtains the form

$$c_v^* C_v \frac{DT}{Dt} = - \nabla \cdot \vec{q} - T \left(\frac{\partial p}{\partial T} \right) \hat{v} \nabla \cdot \vec{v}_A + \Gamma \quad (16)$$

Here

C_v = specific heat of air at constant volume (molar basis)

$\frac{DT}{Dt}$ = substantial derivative of T

$\vec{q}$ = sensible heat flux vector

p = pressure

$\hat{v}$ = specific volume

and Γ = energy generation rate per unit volume from latent-heat effects.

Equation (16) can be simplified appreciably in several respects. From the ideal-gas law the second term on the right becomes

$$c^* RT \nabla \cdot \mathbf{v}_A$$

Applying K theory for heat diffusion, one may write

$$q_z = -c^* C_p D_H \frac{\partial \theta}{\partial z} \quad (17)$$

where C_p is the specific heat at constant pressure (molar basis), $\frac{\partial \theta}{\partial z}$ is the vertical potential temperature gradient, and D_H is the appropriate eddy diffusivity for heat. Finally the mass-continuity equation, in conjunction with the obvious simplifications arising from one-dimensionality, gives

$$w_A \frac{\partial c^*}{\partial z} = -c^* \nabla \cdot \mathbf{v}_A \quad (18)$$

Combining the above features with (16) gives the form

$$\begin{aligned} \frac{DT}{Dt} = & \gamma D_H \frac{\partial^2 \theta}{\partial z^2} + \gamma \left(\frac{\partial D_H}{\partial z} + \frac{D_H}{c^*} \frac{\partial c^*}{\partial z} \right) \frac{\partial \theta}{\partial z} + \frac{R w_A}{c^* C_v} \frac{\partial c^*}{\partial z} T \\ & + \frac{\Gamma}{c^* C_v} \end{aligned} \quad (19)$$

Equation (19) should be considered as a starting point for all modeling evaluations. There are many potential contributors to the latent-heat term Γ (condensation, fusion, sublimation, ...), and it is difficult to provide for any one situation without loss in generality. We shall restrict ourselves at present to systems involving only liquid- and vapor-phase water; thus

$$\Gamma = \lambda r_c + \lambda r_r, \quad (20)$$

where we have chosen (somewhat arbitrarily) to segregate condensation processes, depending upon whether they occur in conjunction with cloud or rain water. Here λ is the latent heat of condensation (molar basis), and r_c and r_r are the molar rates of condensation to cloud and rainwater, respectively. r_c can be

either positive or negative in this example, since cloud droplets can grow by condensation, or shrink via evaporation.

Now if the model cloud is presumed to grow or evaporate in a manner such that equilibrium is maintained exactly, then two situations can occur. If the ambient atmosphere is subsaturated, then no condensation will take place; i.e.,

$$r_c = 0 \quad (21)$$

If on the other hand the atmosphere is saturated with respect to water vapor, then

$$r_c = - \frac{D(\text{SAT})}{Dt} = - \frac{\partial(\text{SAT})}{\partial T} \frac{DT}{Dt} \quad (22)$$

where SAT is the molar concentration of water vapor defined by the saturation curve. Employing this with (19) gives

$$\begin{aligned} \left(1 + \frac{\lambda}{c^* C_v} \frac{\partial \text{SAT}}{\partial T}\right) \frac{DT}{Dt} = \gamma D_H \frac{\partial^2 \theta}{\partial t^2} + \gamma \left(\frac{\partial D_H}{\partial z} + \frac{D_H}{c^*} \frac{\partial c^*}{\partial z} \right) \frac{\partial \theta}{\partial z} \\ + \frac{R w_A}{c^* C_v} \frac{\partial c^*}{\partial z} T + \frac{\lambda r_r}{c^* C_v} \quad (23) \end{aligned}$$

Let $\tau(z)$ represent the temperature experienced by an adiabatic air parcel as it is moved vertically through the atmosphere. From the definition of potential temperature

$$\frac{\partial \theta}{\partial z} = \frac{\partial T}{\partial z} - \frac{\partial \tau}{\partial z} \quad (24)$$

Applying this in equation (24) and expressing in terms of our "terrain-following derivative" yields the form

$$\begin{aligned} (1+\psi) \frac{\hat{DT}}{Dt} = \gamma D_H \frac{\partial^2 T}{\partial z^2} + \left[\gamma \frac{\partial D_H}{\partial z} + \frac{\gamma D_H}{c^*} \frac{\partial c^*}{\partial z} - (1+\psi) w_A' \right] \frac{\partial T}{\partial z} \\ + \frac{R w_A}{c^* C_v} \frac{\partial c^*}{\partial z} T + \frac{\lambda r_r}{c^* C_v} - \gamma \left(D_H \frac{\partial^2 \tau}{\partial z^2} + \frac{D_H}{c^*} \frac{\partial \tau}{\partial z} \frac{\partial c^*}{\partial z} + \frac{\partial \tau}{\partial z} \frac{\partial D_H}{\partial z} \right) \end{aligned}$$

or

$$\frac{\hat{DT}}{Dt} = A1' \frac{\partial^2 T}{\partial z^2} + B1' \frac{\partial T}{\partial z} + C1' T + G' \quad (25)$$

* For present calculations we shall employ the following descriptor:

$$\frac{\partial \tau}{\partial z} = - \frac{m_a g}{C_p + \frac{\lambda}{c^*} \frac{\partial \text{SAT}}{\partial T}}$$

where

$$A1' = \frac{\gamma D_H}{1 + \psi}$$

$$B1' = \frac{\gamma}{1 + \psi} \frac{\partial D_H}{\partial z} + \frac{\gamma D_H}{c^*(1 + \psi)} \frac{\partial c^*}{\partial z} - w_A'$$

$$C1' = \frac{R w_A}{c^* C_V (1 + \psi)} \frac{\partial c^*}{\partial z}$$

$$G' = \left[\frac{\lambda r_r}{c^* C_V} - \gamma \left(D_H \frac{\partial^2 \tau}{\partial z^2} + \frac{D_H}{c^*} \frac{\partial \tau}{\partial z} \frac{\partial c^*}{\partial z} + \frac{\partial \tau}{\partial z} \frac{\partial D_H}{\partial z} \right) \right] / (1 + \psi)$$

and $\psi = 0$ if ambient air is subsaturated

$$= \frac{\lambda}{c^* C_V} \frac{\partial(\text{SAT})}{\partial T} \text{ if ambient air is saturated.}$$

D. Initial and Boundary Conditions for Energy-Balance Relationship

The similarity in form between equations (10) and (25) should be emphasized at this point. This similarity is evident as well in the initial and boundary conditions; the initial condition, for example, is simply

$$T(0, z) = \hat{T}(z) \quad (26)$$

Flux-specified boundary conditions at grid top and bottom are given respectively by

$$\hat{q}_T(t) = w_A' (t, h) T(t, h) c^* C_V - D_H c^* C_p \left(\frac{\partial T}{\partial z} \Big|_h - \frac{\partial \tau}{\partial z} \Big|_h \right) \quad (27)$$

and

$$\hat{q}_B(t) = - D_H c^* C_p \left(\frac{\partial T}{\partial z} \Big|_0 - \frac{\partial \tau}{\partial z} \Big|_0 \right), \quad (28)$$

while their temperature-specified counterparts are

$$T(t, h) = \hat{T}_T(t) \quad (29)$$

and

$$T(t, 0) = \hat{T}_B(t) \quad (30)$$

III. CODE CHARACTERISTICS -- AN OVERVIEW

Listings and a detailed flow diagram of the main program PLUVIUS are provided in Appendix B to this report. It is important at this point, however, to present a brief overview of some essential features of the code, in order to facilitate the following discussion of its application. These features can be consolidated into the three separate areas of 1) initial rules and conventions, 2) data input, and 3) modular construction, and will be discussed individually in the following text.

A. Initial Rules and Conventions

The computer code contains some important conventions, which are designed to maximize flexibility and computational efficiency. These conventions, which must be observed for successful code execution, are itemized in the following paragraphs.

- Temperature is always the first dependent variable to be computed, and is assigned the name $C(I,1)$, where I is the vertical grid index. Temperature may be calculated via finite-difference approximations to equation (25), or it may be stipulated by some other relationship; but a temperature profile is absolutely required before computations of other variables can be performed.
- The code contains two basic types of dependent variables, $C(I,K)$. The first is known as a "computed variable" and is calculated via the code's finite differencing procedure. The second is termed a "constrained variable." This class contains steady-state variables, variables which do not change with time, variables which are force-fit to observations, and generally any variable which is not computed via the finite difference approximations. With the exception of temperature, the ordering of dependent variables is set so that constrained variables follow computed variables. Thus, if $NCOMP$ computed variables and $NCNSTR$ constrained variables exist, the consequence will be

and C(I,1) ... C(I,NCOMP) computed variables
 C(I,NCOMP+1) ... C(I,NCOMP+NCONSTR) constrained variables

If temperature (C(I,1)) is not actually a computed variable, it is still classified as such in the above scheme. The code is informed about the true nature of temperature by the input variable ITEMP:

if ITEMP = 0, C(I,1) is treated as a constrained variable,
if ITEMP = 1, C(I,1) is treated as a computed variable.

- Turbulent diffusivities D(I,LDIF) are grouped to allow the versatility of having different diffusion coefficients for different classes of computed variables, if desired. The index LDIF is known as the "diffusion regime." LDIF = 1 always pertains to the diffusivity of heat in the energy equation (25) regardless of whether or not this equation is actually utilized in the current computation. The highest value of LDIF (total number of separate diffusion regimes) is given by the variable LTD. Normally two sets of diffusivities are required, one for pollutant and one for heat, corresponding to LTD = 2.
- Dimensioning of the variables is rather straightforward, although a few exceptions, related to the diffusion regimes should be noted. These are as follows:

D(I,LTD), DOLD(I,LTD), DCDZ(I,LTD-1), DDDZ(I,LTD-1), DSCDZ (I,LTD-1)

where I must be set to at least the number of vertical grid intersections and LTD must equal or exceed the actual number of diffusion regimes. Dimensioning of the example programs has been purposely set sufficiently high to avoid problems associated with present applications, so that the reader needn't worry about this feature at this point. This should be a first consideration, however, in any extensions of the code.

- Units employed consistently by the code are as follows:

material	gram-moles
force	newtons
length	cm
temperature	°K
time	seconds

- Because of their intrinsic relationship with energy-balance considerations, a semi-rigorous ordering convention has been established with regard to the various forms of atmospheric water. This is a somewhat cumbersome but unavoidable feature of the code, and its discussion can be deferred until a later point. It should be flagged here, however, owing to its intrinsic importance to successful code operations.

B. Data Input

Initial data are read in the main program by a simple sequence of READ statements, as shown in the listing in Appendix B. The first READ statement,

```
      READ (5,10) NCOMP, NCNSTR
10   FORMAT (2I5)
```

identifies the number of species to be considered in the computation, as indicated previously:

NCOMP = number of species who, with the possible exception of ("computed" temperature, will be modeled via finite-difference approximation to equation (10) and its boundary conditions. species)

NCNSTR= number of species, whose concentrations may vary with space ("constrained" and time, but are computed by methods that do not involve species) the finite difference approximations.

The second READ statement,

```
      READ (5,30) (NAME(LL,K), LL=1,5).
30   FORMAT (5A2)
```

provides for input of alphanumeric variable names (e.g. OZONE, SO2).

The third READ statement,

```
      READ (5,100) N, NPRINT,LTD,DZ,T,DT
100  FORMAT (3I5,3E10.3)
```

identifies the following variables:

N	= number of grid points in vertical array,
NPRINT	= number of time steps to be computed before intermediate results are printed,
LTD	= total number of "diffusion regimes",
DZ	= length (in centimeters) of bottom vertical grid element,
T	= model time (in seconds) during which computations are expected to proceed,
DT	= initial time step (in seconds).

The fourth READ statement

```
      READ (5,170)IGEN,IDIFF,ITEMP,IPRESS,ISETTL,IBOUND,IVERT,IFOG
170  FORMAT (8I5),
```

provides for bypassing of a number of subroutines, if their corresponding physical relationships are either constant in time or else unimportant to the modeling problem at hand. If the value of a specific variable is set equal to 1 its corresponding physical phenomenon is subjected to detailed, repetitive calculation and the appropriate subroutines are interrogated at each time step. If a variable is set equal to 0, any related routines are either called only once, during initial stages of the programs execution, or else bypassed completely as appropriate. Significances of the respective variables and their corresponding subroutines are listed in Table 1.

The fifth READ statement

```
40  READ (5,190) ITOP(K), IBOT(K), LDIF(K)
190  FORMAT (3I5),
```

allows the types of boundary conditions to be specified at the tops and bottoms of the vertical grids for each computed species and stipulates its diffusion

TABLE 1: Significances of Bypass Control Variables

<u>Variable</u>	<u>Significance</u>	<u>Corresponding Subroutine</u>
IGEN	physicochemical transformation	GEN
IDIFF	atmospheric mixing	DIFF
ITEMP	temperature	TEMPF
IPRESS	pressure	PRESSF
ISETTL	terminal fall velocity	SETTL
IBOUND	boundary conditions	BOUND
IVERT	vertical air velocity	VERT
IFOG	water condensation	no particular sub- routine; bypasses at several locations throughout code

regime. Setting these equal to 1 imposes flux-specified conditions [cf. equations (12) and (14) or (27) and (28)]. If they are set equal to 0, the boundary conditions become concentration-specified [equations (13) and (15) or (29) and (30)].

These five READ statements complete the necessary input requirements for program execution. Examples of actual input and the corresponding computed output will be given in later sections of this report.

C. Modular Construction

As indicated in the previous text the computer code is organized in a modular form, which is intended to provide the user with a large degree of versatility for specific applications. This subsection presents an overview of this structure by first discussing the modules on an individual basis, and then indicating how they operate as a composite whole. Simple examples of representative subroutines are dispersed periodically throughout the text for purposes of illustration.

Subroutines utilized by this code fall into three general classes, which are described as follows:

Class I Subroutines - Those subroutines which are essential for basic operation of the code, but are intended for modular replacement by the user.

Class II Subroutines - Those subroutines which are intended for utility use within the code and in general should be left unchanged.

Class III Subroutines - Those special-purpose subroutines, to be supplied by the user, which are necessary only for the specific purpose at hand.

Information exchange between the subroutines and the main program is conducted mainly with the use of blank and labeled COMMON. The variable INIT, often appearing by itself in the subroutine argument list, is used as a flag to inform the subroutine whether or not initial grid set-up operations are required. If $INIT \neq 1$ the execution proceeds on the presumption that all initialization has been performed previously, with the result that several non-essential steps are bypassed. Setting of INIT to its appropriate value occurs automatically within the main program.

Since it is necessary that the user be familiar with the Class I subroutines to execute the code, basic aspects of these operations will be discussed at this point. This discussion will be centered around a simple example computer run, whose corresponding input and output will be presented later. The example involves temperature, plus several pollutant species (A-G) as indicated in Table 2. The vertical grid utilized for this situation will consist of twenty equally-spaced elements, extending to a height of 1000 meters, which will be assumed to be the base of a strong, impermeable temperature inversion.

1. *Grid Structure Control*

The code contains three Class I subroutines which are addressed specifically to grid-structure control. Named GRID, TIMESP and VERT, these subroutines control respectively the vertical grid spacing, the temporal grid spacing, and the vertical position of the grid with respect to some previously-defined datum. These are described in the following paragraphs.

SUBROUTINE GRDSET establishes the vertical finite-difference grid as shown in Figure 1. Applied in conjunction with the modified Crank-Nicholson computation scheme described in Appendix A, this grid provides for variable spacing of grid points in the vertical direction. Vertical grid spacing is dictated by a series of expansion factors, $EF(I)$, which are defined by the form

$$DZ(I+1) = DZ(I)*EF(I+1) \quad .$$

Here $DZ(I)$ is the distance between grid indices I and $I+1$, as shown in Figure 1. Subroutine GRDSET creates the $EF(I)$ array and performs all initial housekeeping functions associated with the vertical grid which, once established, remains unchanged throughout program execution. An example listing of GRDSET, pertaining to an equally-spaced grid, is given in Table 3. In general the user should leave the second portion of GRDSET intact, since this is a series of housekeeping operations which must be performed regardless of the nature of $EF(I)$.

As can be noted from Table 3 the vertical grid is controlled essentially by the array $EF(I)$ and the input variables N and DZ , where

N = number of grid intersections in the Z -direction

TABLE 2
Summary of Species Properties
For Simple Example Code Application

<u>SPECIES</u>	<u>REACTION/DEPOSITION BEHAVIOR</u>	<u>INITIAL CONDITIONS</u>
temperature	constrained variable; adiabatic profile fixed in time and space; 288°K at surface	
A	inert	Gaussian*
B	nonreactive, depositing with $v_d = 1$ cm/sec	Gaussian*
C	reactive; $C + D \xrightarrow{k} E$ with $k = 2 \times 10^{12} \frac{\text{cm}^3}{\text{mole sec}}$; non-depositing	Gaussian*
D	reactive; total sink at surface	Gaussian*
E	product; fall velocity = 1 cm/sec, $v_d = 2$ cm/sec	zero
F	<u>constrained variable</u> ; 100 ppb mixing ratio fixed in time and space	
G	<u>constrained variable</u> ; steady state $C + F \xrightleftharpoons{K} G$ with $K = 1 \text{ cm}^3/\text{mole}$	

* Gaussian mixing ratio. Identical initial conditions of $\exp[-(z-h)^2/2\sigma_z^2]$ parts per million will be employed to provide a common comparative basis.

FIGURE 1. Schematic of Computation Grid

TABLE 3

Subroutine GRDSET: Typical Listing

```

SUBROUTINE GRDSET
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),
1DCDZ(50,3),DDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),
2A1(50,4),B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)
DZ1=DZ
NM1=N-1

      FIRST PORTION: SET EF(I) ARRAY

DO 10 I=1,N
EF(I)=1.

      SECOND PORTION: SET UP FACTOR ARRAYS
      NOTE: THE VARIABLE DZ PERTAINS TO DZ(I-1)

Z(1)=0.
DO 20 I=2,NM1
DZ=DZ*EF(I-1)
EFP1(I)=EF(I)+1.
EFSQ(I)=EF(I)*EF(I)
EFSQM1(I)=EFSQ(I)-1.
BETA(I)=EFP1(I)*EF(I)*DZ
GAMMA(I)=BETA(I)*DZ
Z(I)=Z(I-1)+DZ
DZ=DZ*EF(NM1)
Z(N)=Z(N-1)+DZ
DZN=DZ
SIGMA=DZ1*EF(2)*(1+EF(2))
ZETA=(1.+EF(N-1))**2/EF(N-1)**2
EPS=DZN*(1.+1./EF(N-1)-ZETA)
EP1SQ=(1.+EF(2))**2
RETURN
END

```

and DZ = the length in centimeters of the grid space $Z(2) - Z(1)$.*

In Table 3 GRDSET creates a uniform grid of length $DZ*(N-1)$ centimeters, subdivided into $N-1$ elements of DZ centimeters each. To provide a nonuniform grid, one simply changes the subroutine by replacing the DO 10 loop with the desired statements. When implementing a variable configuration the following relationship is sometimes useful:

$$Z(J) = \frac{DZ(1) (1-\epsilon^{J-1})}{1-\epsilon} \quad (31)$$

where ϵ is a constant expansion factor ($EF(I)=\epsilon$ for $I=1, \dots, J$). This expression can be generalized to any consecutive series of constant- ϵ grid points within the total grid.

SUBROUTINE TIMESP defines the length of the time spacing DT at each point of the finite-difference grid. DT is completely variable from one time step to the next. It can be set to a constant value throughout the computation, or it may be adjusted to meet user demands, which may be induced by stiffness phenomena, changes in boundary conditions, or other reasons. A listing of TIMESP, corresponding to a fixed 100-second value of DT , is given in Table 4.

SUBROUTINE VERT dictates the rate of change of the vertical position of the computation grid and the velocity of upward airflow through the grid elements. These are important considerations in computations involving cloud- and precipitation-formation, since such phenomena are usually induced by vertical motions of air masses. VERT simply establishes the variable $DZDT$, which is the time rate of change in grid elevation in cm/sec, and the array $WAP(I)$, which corresponds to w_A' in equation (10).

An example of subroutine VERT is given in Table 5. This corresponds to the somewhat trivial case of zero vertical velocities, which are maintained throughout the computation. If (as is true in this case) $DZDT$ and $WAP(I)$

* DZ is defined as a scalar in the computer program. Its value changes during the initialization process, however, and it is therefore convenient to treat it in terms of a vector ($DZ(I)$, $I=1, \dots, N-1$) in this discussion.

do not change with time, then it is appropriate to suppress subsequent calls of this subroutine by setting the input variable IVERT to zero.

2. *Initial and Boundary Conditions*

SUBROUTINE BOUND computes upper and lower boundary conditions at each time step (if desired) for each of the computed species. If the boundary condition is flux specified FB(K) and/or FT(K) are computed, where FB and FT are the bottom and top vertical fluxes (positive upward) of pollutant in gram-moles/(cm² sec) [cf. equations (12) and (14)]. If the boundary condition is concentration-specified C(1) and/or C(N) are computed, where C is the concentration of species K in gram-moles/cm³ [cf. equations (13) and (15)].

An example subroutine BOUND, corresponding to the variable set in Table 2, is shown in Table 6. The first operation establishes the top fluxes FT=0, corresponding to the impervious layer at 1000 meters. The second sequence of operations establishes the lower boundary conditions at their flux-specified or concentration-specified values. Since temperature [C(I,1)] is a constrained variable in this case, it does not require corresponding boundary conditions to be set. Subsequent interrogations of subroutine BOUND can be suppressed by setting input variable IBOUND to zero; this should be done if and only if none of the boundary conditions need to be updated with time.

SUBROUTINE CONC has two major functions:

- it sets initial conditions for the computer variables
- and
- it computes values for the constrained variables.

Initial conditions are set by loading the initial concentration arrays C(I,K) I=1,N; K=1,NTOT, where N = the number of vertical grid intersections and NTOT is the total number of species in the computation. It should be emphasized at this point that subsequent values of the computed variables will be determined by the finite-difference algorithm. Subsequent constrained-variable values, however, are computed by CONC; thus, when constrained variables are present, CONC must be interrogated at each time step of the computation (ICONC=1).

TABLE 4

Subroutine TIMESP: Typical Listing

```
SUBROUTINE TIMESP
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),
1DCDZ(50,3),DDDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),
2A1(50,4),B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)
BT=100.
RETURN
END
```

TABLE 5

Subroutine VERT: Typical Listing

```
10  SUBROUTINE VERT (T)
      COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
      1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
      2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
      DO 10 I=1,N
      WAP(I)=0.
      DZDT=0.
      RETURN
      END
```

TABLE 6

Subroutine BOUND: Typical Listing

```

SUBROUTINE BOUND (INIT)
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),
1DCDZ(50,3),DDDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),
2A1(50,4),B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)
IF (INIT.NE,1) GO TO 20
DO 10 K=2,NCOMP
10 FT(K)=0.
   FB(2)=0.
   FB(4)=0.
   C(1,5)=0.
20 FB(3)=-1.*C(1,3)
   FB(6)=-2.*C(1,6)
RETURN
END

```

An example subroutine CONC, corresponding to the species set of Table 2, is given in Table 7. Here the DO 10 loop establishes initial concentrations $[C(I,K) \text{ } I=1,N; K=2,NTOT]$ for all species except temperature,^{*} including constrained species. Subsequent calls of CONC update the values of the constrained species as required; these subsequent calls may be bypassed, when appropriate, by setting the bypass variable ICONC to zero. It should be noted here that the vector CS(I) pertains to the concentration of air, in moles/cm³.

3. *Transport Properties*

SUBROUTINE DIFF establishes the vector of vertical diffusion coefficients $D(I,LDIF) \text{ } I=1,N \text{ (cm}^2\text{/sec)}$ at each time step. A simple example of a subroutine defining a fixed array

$$D(I,2) = 100,000 \text{ cm}^2\text{/sec} \quad I=1,N$$

is shown in Table 8. The diffusivity array for heat $D(I,1)$ does not have to be computed for this example, since temperature is a constrained variable. If diffusivity does not change with time, then repeated interrogations of this subroutine should be suppressed by setting IDIFF = 0.

SUBROUTINE SETTLE computes values of w_r as defined in equation (4). Most species of interest will have zero settling velocity, and this must be established in SETTLE by setting elements of the array $W(I,K)$ equal to zero for each of these components at each grid location. Species (such as precipitation water) having significant fall velocities must be accounted by appropriate non-zero entries in the $W(I,K)$ array.

Table 9 provides a simple example of SETTLE. In this example all elements of the array $W(I,K)$ are initially set equal to zero. Species E does have an appreciable fall velocity, however, and is given the vector of values

$$W(I,6) = -1.; \quad I=1,N$$

^{*} Because of thermodynamic requirements, initial values of temperature, $C(I,1) \text{ } I=1,N$, are set in subroutine TEMP, rather than in CONC. Updated values of temperature, if treated as a constrained variable, should be computed in CONC, however. Since temperature does not vary with time in the present example, it does not need updating in this version.

TABLE 7

Subroutine CONC: Typical Listing

```

SUBROUTINE CONC (INIT)
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCUMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),
1LCDZ(50,3),DDDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),
2A1(50,4),B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)
IF (INIT.EQ,0) GO TO 20
DO 10 I=1,N
C(I,2)=1.E-6*CS(I)*EXP(-(Z(I)-50000.):**2/2000000000.)
C(I,3)=C(I,2)
C(I,4)=C(I,2)
C(I,5)=C(I,2)
C(I,6)=0.
C(I,7)=1.E-7*CS(I)
10 C(I,8)=C(I,7)*C(I,4)
GO TO 40
20 DO 30 J=1,N
30 C(I,8)=C(I,7)*C(I,4)
40 RETURN
END

```


TABLE 8

Subroutine DIFF: Typical Listing

```
SUBROUTINE DIFF (INIT)
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSGM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),
1DCDZ(50,3),DDDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),
2A1(50,4),B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)
DO 10 I=1,N
10 D(I,2)=100000.
RETURN
END
```

TABLE 9

Subroutine SETTL: Typical Listing

```

SUBROUTINE SETTL (INIT)
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),
1DCDZ(50,3),DDDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),
2A1(50,4),B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)
IF (INIT.NE,1) GO TO 20
DO 10 I=1,N
DO 10 K=1,NCOMP
10 W(I,K)=0.
20 DO 30 I=1,N
30 W(I,5)=-1
RETURN
END

```

at each time step. One should note here again that all vertical velocities are by convention positive upward; thus a terminal fall velocity will be characterized by a negative value of W. Subsequent calls of SETTLE can be bypassed, when appropriate, by setting ISETTLE=0.

4. *Physical and Chemical Reaction*

SUBROUTINE GEN produces values of the term G, defined in equation (10). In computer-code form this is denoted in terms of the matrix

$$G(I,K); I=1,N; K=1,NCOMP$$

One should note here again that the "constrained" species do not involve solutions of equation (10). Thus GEN has significance only for the "computed" species 1, ... , NCOMP, although constrained variables may be used to determine conversion rates under some circumstances. An example of this is given by G(I,4) in Table 10. As can be noted from Table 2, species numbers 4 and 5 (C) and (D) are determined by both irreversible and equilibrium reactions. The equilibrium term is accounted in Table 10 by the expression

$$C(I,7)*(C(I,4)-COLD(I,4))/DTOLD ,$$

which is an approximation to

$$KF \frac{\partial C}{\partial t} .$$

Physical units of G are gram-moles/cm³ sec, unless specified otherwise by the user.

5. *Thermodynamic Properties*

Temperature and pressure are important thermodynamic properties in many model applications, especially those involving storm phenomena. These properties are specified by the subroutines TEMPF and PRESSF, as described in the following paragraphs.

SUBROUTINE TEMPF, as noted previously, sets up initial conditions for the temperature profile. Under conditions when temperature is treated as a computed variable this routine also performs the vector operations necessary to compute A1', B1', C1' and G' and prepares the system for the Crank-Nicholson integration procedure. Since temperature is a constrained variable in the present appli-

TABLE 10

Subroutine GEN: Typical Listing

```

SUBROUTINE GEN (INIT)
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),
1DCDZ(50,3),DDDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),
2A1(50,4),B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)
IF (INIT.NE.1) GO TO 20
DO 10 I=1,N
COLD(I,4)=C(I,4)
DO 10 K=2,NCOMP
10 G(I,K)=0.
20 DO 30 I=1,N
G(I,4)=-1.E7*C(I,4)*C(I,5)-C(I,7)*(C(I,4)-COLD(I,4))/DTOLD
G(I,5)=G(I,4)
30 G(I,6)=-G(I,4)
RETURN
END

```

cation, only the first function appears in the example listing of Table 11.

SUBROUTINE PRESSF produces a vector of pressure values

PRESS(I) I=1,N (newtons/cm²)

corresponding to the vertical computation grid, for each time step. These are computed on the basis of the hydrostatic equation, and these operations (demonstrated in Table 12) require little elaboration. Repeated calls of PRESSF can be bypassed, when appropriate, by setting the control variable IPRESS equal to zero.

Class II subroutines require little discussion at this point, owing to the fact that they do not in general require user modification. The several Class II subroutines employed by this code are summarized in Table 14. Class III subroutines are totally user-specific and will not be discussed until they are encountered in example applications, which appear in the following sections.

If one now re-examines equations (10)-(15) in the context of the above subroutine descriptions, he can note that all of the parameters in these relationships have been supplied. A1, B1, C1, and G of equation (10) are defined, and all initial and boundary conditions have, in principle, been set. The final remaining task is to obtain the actual finite-difference approximations to the solutions of equation (10), a function which is performed by the composite of the main program and its modular components.

Figure B1 in Appendix B presents a flow chart of the composite program, indicating the linkages between the various modules, and Table 14 provides a complete list of code nomenclature. This chart requires little elaboration; one should note, however, that ultimate set-up of the elements of the finite-difference equations is controlled directly within the main program. This algorithm corresponds directly to the modified Crank-Nicholson scheme which is described in Appendix A.

Subroutine TEMPF: Typical Listing

33

TABLE 12

Subroutine PRESSF: Typical Listing

```

SUBROUTINE PRESSF
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),
1DCDZ(50,3),DDDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),
2A1(50,4),B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)
DO 10 I=1,N
10 PRESS(I)=10.1325*EXP(-1.199E-6*(Z(I)+GLEVL))
RETURN
END

```

TABLE 13

CLASS II SUBROUTINES AND THEIR FUNCTIONS

<u>Subroutine Name</u>	<u>Function</u>
PRINT	Handles all line-printer output from code.
TRIDAG	Solves tridiagonal matrix of simultaneous-equation coefficients arising from finite-difference approximation.
SETVEC	Computes coefficients B1 and C1 for simultaneous equations arising from finite-difference approximation.
DERIV	Computes finite-difference approximations to all derivatives in equation (10), which are not directly dependent upon species concentration.

TABLE 14
SUMMARY OF CODE NOMENCLATURE

<u>Variable Name</u>	<u>Dimensions</u>	<u>Definition</u>
A(I)	-	Coefficients of saturation vapor-pressure curve.
AA(I)	sec ⁻¹	Coefficients in tridiagonal matrix.
A1(I)	cm ² /sec	Coefficient of equations (10 and 10').
ANEW(I)	cm ² /sec	Working array for computation of A1(I).
BB(I)	sec ⁻¹	Coefficients in tridiagonal matrix.
BETA(I)	cm	Grid expansion term $\epsilon(\epsilon+1)\Delta z$.
BNEW(I,K)	cm/sec	Working array for computation of B1(I,K).
B1(I,K)	cm/sec	Coefficient of equations (10 and 10').
C(I,K)	°K or moles/ml	Temperature or species concentration.
CC(I)	sec ⁻¹	Coefficients in tridiagonal matrix.
CFOG	moles/ml	Cloud water concentration.
CNEW(I,K)	sec ⁻¹	Working array for computation of C1(I,K).
COLD(I,K)	°K or moles/ml	Storage array for C(I,K) at previous time step.
CS(I)	moles/ml	Air concentration.
CSUBV	cal/mole °K	Specific heat of air at constant volume.
C1(I,K)	sec ⁻¹	Coefficient of equations (10 and 10').
D(I,LDIF)	cm ² /sec	Eddy diffusion coefficient (mass).
DCDZ(I)	variable	Dummy working array.
DD(I)	moles/ml sec	Companion vector for tridiagonal matrix.
DDDZ(I)	cm/sec	Vertical derivative of eddy diffusivity (mass).
DDHDZ(I)	cm/sec	Vertical derivative of eddy diffusivity (heat).

<u>Variable Name</u>	<u>Dimensions</u>	<u>Definition</u>
DDTDZ(I)	$^{\circ}\text{K}/\text{cm}^2$	Vertical derivative of psuedoadiabatic temperature gradient.
DFACT	$\text{cal}/\text{mole } ^{\circ}\text{K}$	$\frac{\lambda}{c^*} \frac{d(\text{SAT})}{dt}$
DHOLD(I)	cm^2/sec	Values of DH(I) at previous time step.
DP	cm	Equivalent spherical radius of sulfate particle.
DSCDZ(I)	variable	Dummy working array.
DSDT(I)	$\text{moles}/\text{cm}^3 ^{\circ}\text{K}$	$\frac{d(\text{SAT})}{dt}$
DT	sec	Time increment.
DTAUDZ(I)	$^{\circ}\text{K}/\text{cm}$	Psuedoadiabatic temperature gradient.
DZ	cm	Vertical grid step size.
DZDT	cm/sec	Vertical velocity of vertical grid.
DZDUM	cm	Current grid spacing.
DZN	cm	Step size of top grid element.
DZ1	cm	Step size of bottom grid element.
EF(I)	-	Vertical grid expansion factor, ϵ .
EFPI(I)	-	$\epsilon + 1$.
EFSQM1	-	$\epsilon^2 - 1$.
EPS	cm	Dummy variable associated with grid expansion at upper boundary.
EP1SQ	-	Dummy variable associated with grid expansion at lower boundary.
ESAT	mm Hg	Saturation vapor pressure of liquid water.

<u>Variable Name</u>	<u>Dimensions</u>	<u>Definition</u>
FB(K)	moles/cm ² sec or cal/cm ² sec	Vertical flux at lower boundary.
FBOLD(K)	moles/cm ² sec or cal/cm ² sec	Value of FB at previous time step.
FT(K)	moles/cm ² sec or cal/cm ² sec	Vertical flux at upper boundary.
FTOLD(K)	moles/cm ² sec or cal/cm ² sec	Value of FT at previous time step.
G(I,K)	moles/cm ³ sec or °K/cm ³ sec	Generation term in equations (10 and 10').
GAMMA(I)	cm ²	Grid expansion term $\epsilon(\epsilon+1) (\Delta z)^2$
GLEVL	cm	Vertical location of grid bottom with respect to datum.
HEX	moles/l	Hydrogen ion concentration.
I	-	Index (usually associated with position on vertical grid).
IBOT(K)	-	Bottom boundary condition designator: = 0 if concentration or temperature specified; = 1 if flux specified.
IBOUND	-	Bypass variable for boundary-condition sub-routine. Bypass after initial call if IBOUND = 0.
IDIFF	-	Bypass variable for diffusivity subroutine. Bypass after initial call if IDIFF=0.
IDN	-	Index of first equation represented on tridiagonal matrix.
IFLIP	-	Flag controlling initial time averaging of A'l, B'l, and C'l arrays for energy equation.

<u>Variable Name</u>	<u>Dimensions</u>	<u>Definition</u>
IFOG	-	Bypass variable for enforced cloudless conditions; no clouds formed if IFOG=0.
IGEN	-	Bypass variable for generation subroutine. Bypass if IGEN=0.
IHI	-	Print control index.
II, III	-	Dummy indices.
ILO	-	Print control index.
INDEX	-	Flag denoting inclusion of temperature computations in FDA. Index=2 if temperature computations included; =1 if not.
INIT	-	Flag denoting whether initial conditions have been established. INIT=1 prior to setting of initial conditions; =0 afterward.
INT	-	Print control index.
IPRESS	-	Bypass variable for pressure subroutine. Bypass after initial call if IPRESS=0.
IPRINT	-	Print control counter. Subroutine PRINT called whenever IPRINT $\geq$ NPRINT.
ISETTL	-	Bypass variable for terminal fall velocity subroutine. Bypassed after initial call if ISETTL=0.
ISLDER	-	Streamline variable, allowing bypass of subroutine DERIV after initial call, if ISLDER=0.
ISLSET	-	Streamline variable, allowing bypass of subroutine SETVEC after initial call, if ISLSET=0.
ITEMP	-	Bypass variable for energy-balance subroutine. Bypass after initial call if ITEMP=0.
ITOP(K)	-	Top boundary condition designator; =0 if concentration or temperature specified; =1 if flux specified.

<u>Variable Name</u>	<u>Dimensions</u>	<u>Definition</u>
IUP	-	Index of last equation represented on tridiagonal matrix.
IVERT	-	Bypass variable for vertical grid and wind velocity subroutine. Bypass after first call if IVERT=0.
J	-	Print control index.
K	-	Index associated with species.
LDIF(K)	-	Index of diffusion regime.
LGOPT	-	Alphanumeric flag denoting mode of solubility calculators. = Gas for gas-phase concentration supplied; = LIQ for liquid-phase concentration supplied.
LTD	-	Total number of diffusion regimes.
N	-	Number of grid points in vertical array.
NAME(K)	-	Alphanumeric species names.
NCNSTR	-	Number of constrained-variable species in computation.
NCOMP	-	Number of species (including temperature) in computation, which are computed via the finite-difference approximation.
NM1	-	N-1.
NPAGE	-	Page of printed output at current time step.
NTOT	-	Total number of species (including temperature) in computation.
N1, N2	-	Dummy indices.
PRESS(I)	newtons/cm ²	Atmospheric pressure.
R	cal/mole °K	Gas-law constant.
RDT	sec ⁻¹	Reciprocal of time step: 1/Δt.
RSUBR	moles/cm ³ sec	Condensation rate of water vapor on hydrometeors.

<u>Variable Name</u>	<u>Dimensions</u>	<u>Definition</u>
SAT	moles/cm ³	Saturation vapor concentration of liquid water.
SD	-	Dummy variable associated with lower boundary condition for energy balance.
SDOLD	-	Value of SD at previous time step.
SHR	-	Specific heat ratio.
SIGMA	cm	Grid expansion term associated with lower boundary: $(1+\epsilon)\epsilon\Delta Z$.
S1	-	Dummy variable associated with lower boundary conditions.
S1OLD(K)	-	Value of S1 at previous time step.
S2	-	Dummy variable associated with lower boundary conditions.
T	sec	Time limit (model time) for computations.
TC	°C	Temperature.
TCLOCK	sec	Model time at current time step.
TD	-	Dummy variable associated with upper boundary condition for energy balance.
TDOLD	-	Value of TD at previous time step.
TK	°K	Temperature.
T1	-	Dummy variable associated with upper boundary conditions.
T1OLD(K)	-	Value of T1 at previous time step.
T2	-	Dummy variable associated with lower boundary conditions.
UNPSI	-	$1 + \psi$: see equation (25).
V(I)	moles/cm ³ or °K	Solution vector of tridiagonal matrix.

<u>Variable Name</u>	<u>Dimensions</u>	<u>Definition</u>
W(I,K)	cm/sec	Gravitational fall velocity of pollutant species.
WAP(I)	cm/sec	Velocity of vertical air flow past grid elements: W_A' .
W1	sec ⁻¹	Dummy variable used in computing AA, BB, and CC arrays.
W2	sec ⁻¹	Dummy variable used in computing AA, BB, and CC arrays.
Z(I)	cm	Vertical position with respect to the grid base.
ZETA	-	Grid expansion term associated with upper boundary $(1+\epsilon)^2/\epsilon^2$.
ZZ	cm	Height of grid point above datum.

IV. EXAMPLE APPLICATION 1: SIMPLE AIR CHEMISTRY AND DRY DEPOSITION

As noted in the preceding text there are essentially four requirements which must be fulfilled prior to executing the PLUVIUS code for simple applications:

- the main program must be available;
 - all Class II subroutines must be available;
 - all Class I subroutines must be available in appropriate user-modified form;
- and ● appropriate input data must be available.

The first two of the above requirements place no responsibility on the user whatsoever, except for the act of ensuring that both items are present in the object deck. Example listings of the complete ensemble of Class I subroutines, corresponding to the system described in Table 2, have been presented in Tables 3-12. An appropriate set of example input cards, which may be compared with the input statements described in Section III, is shown in Table 15.

The cards in Table 15 are quite straightforward. The first states that there are six "computed" variables and two "constrained" variables in this example. One must note once again that the first variable, temperature, is treated as a computed variable in this declaration, although in actuality it is not. The following eight cards supply the variables' names, while the tenth card informs the system that the vertical grid will consist of 21 grid points, output will be printed at each tenth time step, there are two "diffusion regimes", the length of the lowest grid element is 5.00E3 centimeters, computation will terminate at 7.20E3 seconds of model time, and the initial time step is 1.00E2 seconds in length.

The next card dictates that automatic bypassing will occur for all associated Class I subroutines except GEN and BOUND, which must be interrogated at each time step owing to the variable nature of the chemical-reaction and deposition behavior. The final six cards declare boundary-condition and diffusional status

TABLE 15
Input Data for Example 1

	6	2				
TEMP						
A						
B						
C						
D						
E						
F						
G						
	21	10	2	5.00E3	7.20E3	1.00E2
	1	0	0	0 0	1 0	0
	1	1	1			
	1	1	2			
	1	1	2			
	1	1	2			
	1	0	2			
	1	1	2			

for all computed variables. The first card in this series, pertaining to temperature, is really a dummy in this example, since temperature is treated as a constrained variable. All species observe flux-specified boundary conditions, except for the ground-level condition for species D, which is concentration-specified.

Example output from this code utilization is shown in Tables 16-19. Tables 16 and 17 correspond to initial conditions for the computed and constrained variables, providing temperatures in degrees K and pollutant concentrations in gram-moles per cubic centimeter. The "HEIGHT" column gives the height of each grid intersection above the datum height, which is zero in this case. The four "Gaussian" species (A-D) maximize at 500 meters above ground level; the slight skewing of the distributions with height results from the presentation of these values as concentrations, rather than as mixing ratios. Species F, held fixed at a 100 ppb mixing ratio, shows a similar skewing. Species G concentrations are established simply in terms of the noted equilibrium relationship, $DF=G$.

Distributions of A-D are identical, except for the ground-level element of species D, which has been set equal to 0 as a consequence of the total-retention boundary condition. This numerical similarity of the concentration distributions was chosen intentionally, to aid comparative analysis of the computed results. The initial plume-spread parameter, $\sigma_z = 10000$ cm, was chosen to correspond to 500 seconds of travel time (at the specified diffusivity of $100,000 \text{ cm}^2/\text{sec}$); thus for the nonreactive species, at least, one can visualize the "initial" conditions to occur at a point, 500 seconds downwind from the source, where PLUVIUS "took over" from the analytical solution.

Example output corresponding to 3000 seconds of model time (3500 seconds downwind of the "source") is shown in Tables 18 and 19. Little comment is necessary here except to note that the results present an interesting comparison of behavior in view of differences in stipulated conditions noted in Table 2. Plots corresponding to the data of Table 18 are shown in Figure 2. Here the circular points denote the analytical solution of Heines and Peters (1973), corresponding to the equation

TABLE 16

COMPUTED RESULTS FOR 0.

SECONDS OF MODEL TIME, PAGE 1

CURRENT TIME INCREMENT IS .1000E+03SECONDS

LOWER BOUNDARY POSITION ABOVE DATUM = 0.

CURRENT ASCENT RATE = 0.

HEIGHT	TEMP	A	B	C	D	E
.1000E+06	.278E+03	.145E-15	.145E-15	.145E-15	.145E-15	0.
.9500E+05	.279E+03	.156E-14	.156E-14	.156E-14	.156E-14	0.
.9000E+05	.279E+03	.131E-13	.131E-13	.131E-13	.131E-13	0.
.8500E+05	.280E+03	.860E-13	.860E-13	.860E-13	.860E-13	0.
.8000E+05	.280E+03	.439E-12	.439E-12	.439E-12	.439E-12	0.
.7500E+05	.281E+03	.174E-11	.174E-11	.174E-11	.174E-11	0.
.7000E+05	.281E+03	.539E-11	.539E-11	.539E-11	.539E-11	0.
.6500E+05	.282E+03	.130E-10	.130E-10	.130E-10	.130E-10	0.
.6000E+05	.282E+03	.244E-10	.244E-10	.244E-10	.244E-10	0.
.5500E+05	.283E+03	.356E-10	.356E-10	.356E-10	.356E-10	0.
.5000E+05	.283E+03	.405E-10	.405E-10	.405E-10	.405E-10	0.
.4500E+05	.284E+03	.359E-10	.359E-10	.359E-10	.359E-10	0.
.4000E+05	.284E+03	.248E-10	.248E-10	.248E-10	.248E-10	0.
.3500E+05	.285E+03	.133E-10	.133E-10	.133E-10	.133E-10	0.
.3000E+05	.285E+03	.558E-11	.558E-11	.558E-11	.558E-11	0.
.2500E+05	.286E+03	.182E-11	.182E-11	.182E-11	.182E-11	0.
.2000E+05	.286E+03	.462E-12	.462E-12	.462E-12	.462E-12	0.
.1500E+05	.287E+03	.914E-13	.914E-13	.914E-13	.914E-13	0.
.1000E+05	.287E+03	.141E-13	.141E-13	.141E-13	.141E-13	0.
.5000E+04	.288E+03	.169E-14	.169E-14	.169E-14	.169E-14	0.
0.	.288E+03	.153E-15	.158E-15	.158E-15	0.	0.

TABLE 17

CONSTRAINED RESULTS FOR 0,
CURRENT TIME INCREMENT IS .1000E+03SECONDS

SECONDS OF MODEL TIME, PAGE 1

LOWER BOUNDARY POSITION ABOVE DATUM = 0,

CURRENT ASCENT RATE = 0.

HEIGHT	F	G
.1000E+06	.388E-11	.562E-27
.9500E+05	.390E-11	.610E-26
.9000E+05	.392E-11	.515E-25
.8500E+05	.393E-11	.338E-24
.8000E+05	.395E-11	.173E-23
.7500E+05	.397E-11	.692E-23
.7000E+05	.398E-11	.215E-22
.6500E+05	.400E-11	.520E-22
.6000E+05	.402E-11	.979E-22
.5500E+05	.404E-11	.144E-21
.5000E+05	.405E-11	.164E-21
.4500E+05	.407E-11	.146E-21
.4000E+05	.409E-11	.101E-21
.3500E+05	.411E-11	.547E-22
.3000E+05	.412E-11	.230E-22
.2500E+05	.414E-11	.753E-23
.2000E+05	.416E-11	.192E-23
.1500E+05	.418E-11	.381E-24
.1000E+05	.419E-11	.590E-25
.5000E+04	.421E-11	.711E-26
0.	.423E-11	.667E-27

TABLE 18

COMPUTED RESULTS FOR ,3000E+04 SECONDS OF MODEL TIME, PAGE 1
 CURRENT TIME INCREMENT IS ,1000E+03SECONDS

LOWER BOUNDARY POSITION ABOVE DATUM = 0, CURRENT ASCENT RATE = 0.

HEIGHT	TEMP	A	B	C	D	E
,1000E+06	,278E+03	,496E-11	,496E-11	,381E-11	,381E-11	,783E-12
,9500E+05	,279E+03	,520E-11	,520E-11	,396E-11	,396E-11	,952E-12
,9000E+05	,279E+03	,589E-11	,589E-11	,440E-11	,440E-11	,124E-11
,8500E+05	,280E+03	,698E-11	,698E-11	,506E-11	,506E-11	,166E-11
,8000E+05	,280E+03	,836E-11	,836E-11	,589E-11	,589E-11	,221E-11
,7500E+05	,281E+03	,994E-11	,994E-11	,681E-11	,681E-11	,286E-11
,7000E+05	,281E+03	,116E-10	,116E-10	,773E-11	,773E-11	,357E-11
,6500E+05	,282E+03	,131E-10	,131E-10	,856E-11	,855E-11	,427E-11
,6000E+05	,282E+03	,143E-10	,143E-10	,922E-11	,921E-11	,488E-11
,5500E+05	,283E+03	,151E-10	,151E-10	,965E-11	,964E-11	,533E-11
,5000E+05	,283E+03	,154E-10	,154E-10	,981E-11	,979E-11	,556E-11
,4500E+05	,284E+03	,152E-10	,152E-10	,969E-11	,964E-11	,554E-11
,4000E+05	,284E+03	,144E-10	,144E-10	,930E-11	,921E-11	,527E-11
,3500E+05	,285E+03	,133E-10	,132E-10	,868E-11	,851E-11	,478E-11
,3000E+05	,285E+03	,118E-10	,118E-10	,788E-11	,759E-11	,416E-11
,2500E+05	,286E+03	,102E-10	,102E-10	,699E-11	,649E-11	,346E-11
,2000E+05	,286E+03	,862E-11	,855E-11	,609E-11	,527E-11	,278E-11
,1500E+05	,287E+03	,723E-11	,711E-11	,528E-11	,399E-11	,217E-11
,1000E+05	,287E+03	,615E-11	,595E-11	,464E-11	,267E-11	,168E-11
,5000E+04	,288E+03	,547E-11	,514E-11	,423E-11	,134E-11	,136E-11
0.	,288E+03	,525E-11	,473E-11	,411E-11	0.	,121E-11

TABLE 19

CONSTRAINED RESULTS FOR ,3000E+04 SECONDS OF MODEL TIME, PAGE 1
 CURRENT TIME INCREMENT IS .1000E+03SECONDS

LOWER BOUNDARY POSITION ABOVE DATUM = 0, CURRENT ASCENT RATE = 0.

HEIGHT	F	G
.1000E+06	.388E-11	.148E-22
.9500E+05	.390E-11	.155E-22
.9000E+05	.392E-11	.172E-22
.8500E+05	.393E-11	.199E-22
.8000E+05	.395E-11	.233E-22
.7500E+05	.397E-11	.270E-22
.7000E+05	.398E-11	.308E-22
.6500E+05	.400E-11	.342E-22
.6000E+05	.402E-11	.370E-22
.5500E+05	.404E-11	.389E-22
.5000E+05	.405E-11	.398E-22
.4500E+05	.407E-11	.395E-22
.4000E+05	.409E-11	.380E-22
.3500E+05	.411E-11	.356E-22
.3000E+05	.412E-11	.325E-22
.2500E+05	.414E-11	.289E-22
.2000E+05	.416E-11	.253E-22
.1500E+05	.418E-11	.220E-22
.1000E+05	.419E-11	.194E-22
.5000E+04	.421E-11	.178E-22
0.	.423E-11	.174E-22

$$\begin{aligned}
c = & \frac{\text{constant } c^*}{u \sqrt{D_z t}} \sum_{j=0}^{\infty} \left\{ \exp \left[\frac{-[(2j + 1/2) H^* - z]^2}{4D_z t} \right] + \right. \\
& \exp \left[\frac{-[(2j + 1/2) H^* + z]^2}{4D_z t} \right] + \exp \left[\frac{-[(2j + 3/2) H^* + z]^2}{4D_z t} \right] \\
& \left. + \exp \left[\frac{-[(2j + 3/2) H^* - z]^2}{4D_z t} \right] \right\}
\end{aligned}$$

where H^* is the mixed-layer height and t denotes travel time from the source.

This example run was continued to two hours of simulation time. Corresponding execution time was 1.8 seconds on the Battelle CDC 6400 computer system, under standard FORTRAN compiler mode.

FIGURE 2. Results of Example 1 Computations at 3000 Seconds of Simulation Time From Initial State

V. EXAMPLE APPLICATION 2: REACTIVE SCAVENGING IN AN OROGRAPHIC STORM

A. General Description

The second example application of PLUVIUS addresses the precipitation scavenging of aerosols and reactive gases. In this capacity the code is required to simulate the storm in which the scavenging processes occur; and this in turn requires the additional considerations of two-phase systems, evaporation-condensation effects, and a much more elaborate description of the energy-exchange process.

The basic system selected for this illustration is shown schematically in Figures 3 and 4. As was the case with the preceding example, this system was chosen more for clarity of illustration than for physical rigor; it consists of a rather simple material balance for atmospheric water, and a limited mechanism for capture and interconversion of SO_2 and sulfate. As can be noted from Figure 3, only three classes of atmospheric water are allowed. Ice-formation processes are not included with this example, and cloud and precipitation water are lumped into two respective classes. Once water-saturation conditions exist, clouds form to maintain the system precisely on the saturation curve. Precipitation-sized drops form in this example both by coagulation of cloud droplets, and accretion of droplets by falling rain.

The sulfur balance of Figure 4 allows two basic mechanisms of sulfur scavenging. The first of these is the direct uptake of sulfate aerosol via the cloud-droplet nucleation process. The second concerns the uptake of gas-phase SO_2 by existing cloud droplets with subsequent oxidation via the mechanism

SO_2 and ozone concentrations in cloud water are determined totally on the basis of solubility equilibrium. SO_2 solubility is both temperature and pH dependent. Hydrogen-ion concentrations for these computations are assumed, again for simplicity, to equal $10^{-5} + 0.5$ times the sulfate concentrations in moles per liter. Sulfate aerosol is assumed to be monodisperse, with a particle

Figure 3
Interconversion Diagram for Water

Figure 4

Interconversion Diagram for Sulfur

size which can vary parametrically. When saturation occurs, a set number concentration of cloud droplets forms; this depletes an equal concentration of aerosol particles, which are presumed to act as nuclei. New sulfate particles are allowed to become active nuclei as cloud droplets are removed, until the local air is completely scavenged of its aerosol.

One should note from Figures 3 and 4 that this example's bookkeeping arrangement allows for the individual treatment of primary and secondary sulfate; these are actually treated as separate variables throughout the computation, thus allowing an assessment of relative importance of the two mechanisms.

Mathematical specifications for the interconversion processes depicted schematically in Figures 3 and 4 are given in Table 20. Little needs to be said in elaboration of this table except to note once again that this is a highly lumped depiction; much more elaborate schemes could be incorporated within the PLUVIUS framework, if desired by the user.

B. Ordering of Variables

As noted in Section III.A, the precise ordering of dependent variables within PLUVIUS is arbitrary, so long as a few initial conventions are maintained. The ordering chosen for present purposes is shown in Table 21, which defines also the computational natures of the variables. Some of the computed variables (eg. cloud and water vapor) are actually combinations of computed and constrained species, and were selected purely on the basis of computational convenience. It is important to note that in contrast to example 1, temperature is a computed variable in the present case.

It should be noted as well that species 2 and 3 in this example pertain to cloud and water vapor, and rain, respectively. This is an important consideration because of the energy-balance relationships; and the main program and subroutine TEMPF presented in Appendix B and Table 29 have been written using this "semi-rigorous" convention. This convention can be changed by the user, if desired, and the pertinent Fortran statements in PLUVIUS and TEMPF have been flagged by lines of # symbols for this purpose. Under conditions where cloud-formation computations are bypassed (IFOG=0, eg. Example Application 1) this ordering is immaterial.

TABLE 20
SUMMARY OF PARAMETERIZATIONS USED FOR EXAMPLE COMPUTATION*

<u>Step</u>	<u>Type</u>	<u>Means of Parameterization</u>	<u>Reference</u>
W1	equilibrium	mathematical fit to saturation curve	Pruppacher and Klett (1978)
W2	kinetic	$r_{W2} = 7,580 [\text{cloud}] [\text{rain}]^{.875};$	Scott (1981)
W3	kinetic	$r_{W3} = 600 [\text{cloud}] [\text{cloud}];$	Berry and Reinhardt (1974)
W4	kinetic	$r_{W4} = 28.3 [\text{saturation deficit}] [\text{rain}]^{.65};$	Kessler (1968)
W5	dynamic	$w = -1.45 \text{ E}+11 [\text{rain}]^{1.2}$	Kessler (1968)
S1	equilibrium	mathematical fit to saturation curve	Hales and Sutter (1973)
S2	equilibrium	$[O_3]_l = .45 [O_3]_g$	Kirk and Othmer (1947)
S3	kinetic	$r_3 = \frac{2900}{\sqrt{[H^+]}} [O_3]_l [SO_2]_l$	Penkett, et al. (1979)
S4	constrained to cloud-droplet population	population = 300 cm^{-3}	Mason (1971)

* Units are gram-moles, centimeters, and seconds.

Bracketed terms, eg. [cloud], [rain], [SO₂] signify concentrations of the specified components in gram-moles per ml total volume, with the exception of [H⁺], which is molar.

TABLE 21
Ordering and Initial Conditions for Example Application 2

<u>Variable</u>	<u>Type</u>	<u>Initial Conditions</u>
temperature	computed	psuedoadiabatic, with $C(1,I)=288.15K$
vapor and cloud water	computed	$c(z) = 3.7 \times 10^{-7} \exp (2.93 \times 10^{-6} z)$
rain	computed	0
rainborne secondary sulfate	computed	0
rainborne primary sulfate	computed	0
secondary sulfate in cloud water	computed	0
gas-phase SO_2	computed	$c(z) = 5.1 \times 10^{-13} r(z)$
primary sulfate in air and cloud water	computed	$c(z) = 5.1 \times 10^{-14} r(z)$
primary sulfate in cloud water	computed	0
cloud water	constrained	saturation curve
SO_2 in cloud water	constrained	solubility equilibrium
gas-phase ozone	constrained	$c(z) = 2.2 \times 10^{-12} r(z)$

$r(z) = c^*(z)/c^*(0)$. All concentrations are in gram-moles/cm³.

C. Grid Structure and Dynamic Forcing

The storm simulated by this example is an orographic disturbance which is induced by forcing the initial air column over the simple plateau shown in Figure 5. The vertical grid is 3000 meters high and is composed of twenty equally-spaced segments. Because of this equal spacing the example subroutine GRDSET in Table 3 will apply directly to the present case. Vertical movement of the air column is simulated in the subroutine VERT by adjusting the variable DZDT in the following manner:

DZDT = 0 for TCLOCK $\leq$ 6000 seconds

DZDT = 10 cm/sec for 6000 < TCLOCK $\leq$ 12000 seconds

DZDT = 0 for 12000 < TCLOCK $\leq$ 18000 seconds

DZDT = 10 cm/sec for TCLOCK > 18000 seconds,

where the variable TCLOCK denotes simulation time.

No vertical flow of air through the grid is allowed in this example. The corresponding subroutine VERT is shown in Table 22.

In contrast to Example Application 1, temporal grid spacing is adjusted in the present application. This becomes necessary here because the terms

$$c \frac{\partial w_r}{\partial z} \quad \text{and} \quad w_r \frac{\partial c}{\partial z}$$

of Equation (10) can become quite large during rainfall situations, and tend to create instabilities in the numerical approximation. This undesirable behavior is controlled in the current example by adjusting the time steps so that the concentration changes corresponding to the above terms are less than one tenth of the existing concentration. Upper and lower limits to the time steps are set at 300 and 5 seconds, respectively. The corresponding subroutine TIMESP is given in Table 23.

FIGURE 5. Outlines of Cloud and Precipitation Regions for Model Storm

TABLE 22

Subroutine VERT: Listing for Example 2

```

SUBROUTINE VERT (T)
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
IF (T.GT.6000.) GO TO 20
DO 10 I=1,N
10  WAP(I)=0.
    DZDT=0.
    GO TO 50
20  IF (T.GT.12000.) GO TO 30
    DZDT=10.
    GLEVL=GLEVL+DZDT*DT
    GO TO 50
30  IF (T.GT.18000.) GO TO 40
    DZDT=0.
    GO TO 50
40  DZDT=-10.
    GLEVL=GLEVL+DZDT*DT
50  RETURN
END

```

TABLE 23

Subroutine TIMESP: Listing for Example 2

```

SUBROUTINE TIMESP
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ,
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),D
1CDZ(50,3),DDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),A1(50,4),
2B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)
NM1=N-1
D4=999999999999.
D3=999999999999.
DNEW=9999999999.
DO 30 I=1,NM1
  IF (C(I,3).LT.1.E-10) GO TO 30
  DZSTAR=Z(I+1)-Z(I)
  DCDZ1=(C(I+1,3)-C(I,3))/DZSTAR
  DWDZ=(W(I+1,3)-W(I,3))/DZSTAR
  D1=ABS(W(I,3)*DCDZ1)
  D2=ABS(C(I,3)*DWDZ)
  IF (D1.LE.0.) GO TO 10
  D3=C(I,3)/D1
10  IF (D2.LE.0.) GO TO 20
  D4=C(I,3)/D2
20  IF (D4.LT.DNEW) DNEW=D4
  IF (D3.LT.DNEW) DNEW=D3
30  CONTINUE
  DT=.1*DNEW
  IF (DT.GT.300.) DT=300.
  IF (DT.LT.5.) DT=5.
  RETURN
END

```

D. Initial and Boundary Conditions

The grid top for this example was stipulated to define the bottom of an impervious layer; thus top boundary conditions for all computed species are flux-specified, with the flux set equal to zero. Bottom boundary conditions are set in a similar fashion, except for rain and rainborne sulfate. Because of their associated downward fall velocities these components can have significant surface fluxes, and create an "outflow" boundary situation which leads to some difficulty in exact specification. This problem is countered in the present example by treating the boundary conditions as concentration-specified, and *predicting* the surface concentrations at the current time step on the basis of previous information using a forward differencing procedure. The subroutine BOUND that accomplishes these operations is shown in Table 24.

Initial conditions for Example Application 2 are shown in Table 21. These were chosen to represent reasonably typical polluted atmospheric conditions, with sufficient moisture to create light cloud cover aloft (cf. Figure 5). Equilibrium conditions for dissolved SO₂ and cloud water are dictated by Class III subroutines, which will be described later. Ozone concentrations are constrained to remain fixed in time for the purposes of this calculation. The corresponding listing of subroutine CONC is given in Table 25.

E. Transport Properties

The differences in terminal fall velocities between raindrops in a size-distributed system results in a dispersion of precipitation, which has been parameterized in the present example in terms of a pseudo diffusion coefficient. This parameterization, which presupposes that the raindrop spectrum obeys a Marshall-Palmer distribution, is based upon a moment analysis of an ensemble of falling raindrops. Its form is

$$D_{\text{rain}} = 100,000 + 3.98 \times 10^{10} \Delta t [\text{rain}]^{0.92} \text{ cm}^2/\text{sec} \quad (33)$$

where [rain] denotes the local rainwater concentration in moles/cm³, and Δt is the current time step in seconds. Effective dispersion coefficients for rainborne sulfate as well as for rain are computed for this example using Equation (33).

TABLE 24

Subroutine BOUND: Listing for Example 2

```

SUBROUTINE BOUND (INIT)
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),D
1CDZ(50,3),DDDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),A1(50,4),
2B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)
IF (INIT.NE.1) GO TO 20
ESQ=EFP1(2)*EFP1(2)
DZSQ=DZ1*DZ1
DO 10 K=1,NCOMP
FT(K)=0.
10 FB(K)=0.
20 DDDZ1=(D(1,3)*(1.-ESQ)+D(2,3)*ESQ-D(3,3))/BETA(2)
DWDZ=(W(1,3)*(1.-ESQ)+W(2,3)*ESQ-W(3,3))/BETA(2)
DO 30 J=3,5
TERM1=(C(3,J)/(EF(2)*EFP1(2))-C(2,J)/EF(2)+C(1,J)/EFP1(2))*2.*D(1,
13)/DZSQ
TERM2=(C(1,J)*(1.-ESQ)+C(2,J)*ESQ-C(3,J))*(DDDZ1-W(2,3))/BETA(2)
30 C(1,J)=(TERM1+TERM2-DWDZ*C(1,J)+G(1,J))*DT+C(1,J)
RETURN
END

```

TABLE 25

Subroutine CONC: Listing for Example 2

```

SUBROUTINE CONC (INIT)
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),D
1CDZ(50,3),DDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),A1(50,4),
2B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)

IF (INIT.EQ.0) GO TO 30
NTOT=NCOMP+NCNSTR
DP=1.E-5
HEX=1.E-5
LGOPT=3HGAS
XNC=300.
DPCUBE=DP*DP*DP
NM1=N-1
DUM=.00524*DPCUBE
DO 10 K=2,NTOT
DO 10 I=1,N
10 C(I,K)=0.
DO 20 I=1,N
RATIO=CS(I)/CS(1)
C(I,8)=5.1E-14*RATIO
C(I,2)=.370E-6*EXP(-.293E-5*Z(I))
C(I,7)=5.1E-13*RATIO
20 C(I,12)=2.2E-12*RATIO
30 DO 90 I=1,N
CS(I)=1.203E-3*PRESS(I)/C(I,1)
CFOG=C(I,2)-SAT(C(I,1))
IF (CFOG.LT.0.) GO TO 40
C(I,10)=CFOG
HEX=1.E-5+27.78*(C(I,6)+C(I,9))/C(I,10)
GO TO 50
C      NO CLOUD CASE
40 C(I,10)=0.
C      CONVERT SECONDARY SO4 FROM ANY EVAPORATING DROPS
C      TO PRIMARY SO4
C(I,8)=C(I,8)+C(I,6)
C(I,6)=0.
C(I,9)=0.
C(I,11)=0.
GO TO 80

```

Table 25 Cont.

```

C      CALCULATE NUMBER OF SO4 PARTICLES AND COMPARE WITH
C      NUMBER OF CLOUD DROPS
50     XNP=C(I,8)/DUM
      IF (XNP.LT.XNC) GO TO 60
C      CLOUD WATER SULFATE LIMITED BY NUMBER OF CLOUD DROPS
      C(I,9)=DUM*XNC
      GO TO 70
C      CLOUD WATER SULFATE LIMITED BY NUMBER OF SO4 PARTICLES
60     C(I,9)=C(I,8)
C      SO2 ABSORPTION EQUILIBRIUM
70     P=PRESS(I)/10.1325
      CMF=C(I,7)/CS(I)
      CALL HENRY (HPRIME,CMF,HEX,C(I,1),P,LGOPT)
      HPRIME=HPRIME*CS(I)
      C(I,11)=C(I,7)*C(I,10)/HPRIME
C      ULTIMATE EVAPORATION OF RAIN
80     IF (C(I,3).LT.0.) C(I,3)=0.
      IF (C(I,3).GT.1.E-11.OR.CFOG.GE.0.) GO TO 90
      C(I,2)=C(I,2)+C(I,3)
      C(I,3)=0.
      C(I,8)=C(I,4)+C(I,5)+C(I,8)
      C(I,4)=0.
      C(I,5)=0.
90     CONTINUE
      RETURN
      END

```

Eddy diffusion coefficients for heat and airborne pollutant are set at $100,000 \text{ cm}^2/\text{sec}$, similar to Example 1. Thus three diffusivity regimes exist for this case, given by

$$D(I,1) = 100,000.$$

$$D(I,2) = 100,000.$$

$$D(I,3) = 100,000 + 3.98 \times 10^{10} \text{ At } c(I,3)^{0.92}.$$

The corresponding subroutine DIFF is given in Table 26.

The listing of subroutine SETTLE employed for this example is given in Table 27. Here the fall velocities of all species are set equal to zero, except for those associated with precipitation. These exceptions include rain and rainborne primary and secondary sulfate (species 3, 4, and 5); and their respective fall velocities are parameterized in the subroutine as stipulated in Table 20.

F. Physical and Chemical Reaction

The listing of subroutine GEN employed for this example is presented in Table 28. Here the rate equations have been consolidated somewhat to increase computational efficiency, but reflect directly the parameterizations of Table 20.

Updating of constrained variables is treated in subroutine CONC by the block of Fortran statements starting with statement 30. One should note that some modification of *computed* variables occurs in this block as well. This practice is often convenient when a computed variable depends partially on the behavior of a constrained variable, or some more-or-less contrived interchange mechanism (such as the nucleation mechanism employed here) is utilized. One should note particularly in this context that the variable DP corresponds to the equivalent diameter of a spherical sulfate particle. Set equal to 10^{-5} cm in Table 25, this parameter could be modified easily to perform computations for aerosols having other particle sizes.

G. Thermodynamic Properties

The vertical pressure distribution subroutine for Example 2 is identical to that shown in Table 12 for Example 1. The temperature-distribution subroutine,

TABLE 26

Subroutine DIFF: Listing for Example 2

```

SUBROUTINE DIFF (INIT)
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DCOLD(50,4),W(50,25),D
1CDZ(50,3),DDDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),A1(50,4),
2B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)
DO 10 I=1,N
D(I,2)=100000.
D(I,1)=D(I,2)
D(I,3)=D(I,2)
IF (INIT.EQ.1) GO TO 10
D(I,3)=D(I,2)+3.98E10*DT*C(I,3)**.92
10 CONTINUE
RETURN
END

```

TABLE 27

Subroutine SETTL: Listing for Example 2

```

SUBROUTINE SETTL (INIT)
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),D
1CDZ(50,3),DDDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),A1(50,4),
2B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)
IF (INIT.NE.1) GO TO 20
NDUM=N-1
DO 10 I=1,N
DO 10 K=1,NCOMP
10 W(I,K)=0.
20 DO 30 I=1,N
W(I,3)=-1.45E11*C(I,3)**1.2
W(I,5)=W(I,3)
30 W(I,4)=W(I,3)
RETURN
END

```

TABLE 28

Subroutine GEN: Listing for Example 2

```

SUBROUTINE GEN (INIT)
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),D
1CDZ(50,3),DDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),A1(50,4),
2B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)
DIMENSION RK(4)
DATA RK/3.06E8,2.00E3,7.58E3,28.3/
C   RK(1) = OZONE OXIDATION COEFFICIENT
C   RK(2) = CLOUD WATER COAGULATION COEFFICIENT
C   RK(3) = ACCRETION COEFFICIENT
C   RK(4) = RAINDROP EVAPORATION COEFFICIENT
DO 20 I=1,N
  IF (C(I,10).LE.0.) GO TO 10
  HEX=1.0E-5+27.78*(C(I,6)+C(I,9))/C(I,10)
  RK(1)=2.9E6/HEX**.5
10  D1=-RK(2)*C(I,10)-RK(3)*C(I,3)**.875
  D2=-RK(4)*(C(I,2)-C(I,10)-SAT(C(I,1)))
C   RATE OF PRIMARY SULFATE TRANSFER TO RAIN
  G(I,8)=D1*C(I,9)
C   RATE OF PRIMARY SULFATE GAIN BY RAIN
  G(I,5)=-G(I,8)
C   RATE OF GAIN OF SECONDARY SULFATE IN CLOUD WATER
  D11=RK(1)*C(I,12)*C(I,11)
  G(I,6)=D11+D1*C(I,6)
C   RATE OF PICKUP OF SECONDARY SULFATE BY RAIN
  G(I,4)=-D1*C(I,6)
C   RATE OF GAIN OF VAPOR AND CLOUD WATER
  G(I,2)=D1*C(I,10)+D2*C(I,3)**.65
C   RATE OF RAIN GENERATION
  G(I,3)=-G(I,2)
C   RATE OF SO2 GENERATION
20  G(I,7)=-D11
  RETURN
END

```

however, is radically different. Shown in Table 29, this subroutine computes an initial temperature distribution on its first interrogation, and generates A1', B1', C1', and G' arrays for the finite difference approximation to Equation (25) on all subsequent calls.

Evaporation and condensation effects on the energy balance are treated by computing equilibrium saturation concentrations (in the case of cloud water) and evaporation rates (in the case of rain). Two Class III subroutines SAT and DSAT are used to aid these operations. Subroutine SAT simply computes the equilibrium concentration of water vapor in moles/cm³ at a given temperature T. DSAT provides the corresponding temperature derivative, in moles/(cm³°K). These routines will be discussed in more detail in the following section. The need for subroutine DSAT can be noted from Equation (25) and also from the energy balance used for establishment of the initial conditions:

$$\frac{\partial T}{\partial z} = \frac{-m_a g}{c_p + \lambda/c^* \frac{\partial SAT}{\partial T}} \quad , \quad (33)$$

where m_a is the molecular weight of air and g is the gravitational acceleration.

One should note that this particular version of subroutine TEMPF can be adapted easily to a wide variety of situations with some relatively simple modifications. If an ice phase were to be included, for example, one would simply substitute appropriate statements for those flagged by the # signs in the listing. Again, adherence to a "semirigorous" ordering convention for the classes of water is mandatory.

H. Class III Subroutines

As noted above Example Application 2 requires the incorporation of several special-purpose, or "Class III" subroutines. The first two of these, SAT and DSAT, have been mentioned in the previous text and are presented in Table 30. Both SAT and DSAT are simple applications of textbook formulae [Pruppacher and Klett (1978)] and require little elaboration at this point. The single argument is temperature in degrees Kelvin, and the return-variable units are moles/cm³ and moles/(cm³°K), respectively.

TABLE 29

Subroutine TEMPF: Listing for Example 2

```

SUBROUTINE TEMPF (INIT,IFOG)
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),D
1CDZ(50,3),DDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),A1(50,4),
2B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)
DIMENSION DDHDZ(50), DSDT(50), ANEW(50), DTDZ(50), DDTDZ(50)
*****
C
C      PASS1 BLOCK: COMPUTES INITIAL TEMPERATURE, CS, AND
C      DIFFUSIVITY PROFILES, AND INITIALIZES VARIABLES.
C      SUBSEQUENT CS PROFILES ARE COMPUTED IN SUBROUTINE
C      CONC.
C
C      *****
C      IF (INIT.NE.1) GO TO 30
C      NM1=N-1
C      C(1,1)=288.15
C      CS(1)=1.203E-3*PRESS(1)/C(1,1)
C      DO 20 I=2,N
C      DZDUM=Z(I)-Z(I-1)
C      DFACT=0.
C
C      C#####
C
C      C(I-1,2)=.370E-6*EXP(-.293E-5*Z(I))
C      IF (IFOG.EQ.0) GO TO 10
C      CFOG=C(I-1,2)-SAT(C(I-1,1))
C
C      C#####
C
C      IF (CFOG.LT.0.) GO TO 10
C      DFACT=10800.*DSAT(C(I-1,1))/CS(I-1)
10  C(I,1)=C(I-1,1)-DZDUM*.000701/(7.176+DFACT)
20  CS(I)=1.203E-3*PRESS(I)/C(I,1)
C      R=1.987
C      SHR=1.3829
C      CSURV=5.189
C      IFLIP=0
C

```

Table 29 Cont.

```

C *****
C
C PASS 2 .... BLOCK: COMPUTES A1,B1,C1, AND G ARRAYS
C FOR FINITE DIFFERENCE APPROXIMATION.
C *****
C
C GO TO 120
30 IF (IFLIP.EQ.0) GO TO 50
C
C IFLIP CONTROLS STEPWISE AVERAGING OF THE A,B,
C AND C ARRAYS. ON FIRST EXERCISE OF THE F.D.A.
C NO AVERAGING IS PERFORMED, ON SUBSEQUENT STEPS
C THE OLD AND NEW VALUES ARE AVERAGED.
C
C DO 40 I=1,N
C CS(I)=1.203E-3*PRESS(I)/C(1,1)
C A1(I,1)=ANEW(I)
C B1(I)=BNEW(I,1)
40 C1(I)=CNEW(I,1)
C
C COMPUTE DSAT/DT
C
50 DO 60 I=1,N
C DFACT=0.
C IF (IFOG.EQ.0) GO TO 60
C DSDT(I)=0.
C
C #####
C
C CFOG=C(I,2)-SAT(C(I,1))
C
C #####
C
C IF (CFOG.LT.0.) GO TO 60
C DSDT(I)=DSAT(C(I,1))
C DFACT=10800.*DSDT(I)/CS(I)
60 DTDZ(I)=-.000701/(7.176+DFACT)
C
C COMPUTE DDHDZ AND DCDZ
C

```

Table 29 Cont.

```

DO 70 I=2,NM1
DDHDZ(I)=(D(I+1,1)-EFSQ(I)*D(I-1,1)+EFSQM1(I)*D(I,1))/BETA(I)
DDTDZ(I)=(DTDZ(I+1)-EFSQ(I)*DTDZ(I-1)+EFSQM1(I)*DTDZ(I))/BETA(I)
70 DCDZ(I,1)=(CS(I+1)-EFSQ(I)*CS(I-1)+EFSQM1(I)*CS(I))/BETA(I)
DCDZ(1,1)=(CS(1)*(1.-EP1SQ)+CS(2)*EP1SQ-CS(3))/SIGMA
DDTDZ(1)=(DTDZ(1)*(1.-EP1SQ)+DTDZ(2)*EP1SQ-DDTDZ(3))/SIGMA
DDTDZ(N)=(ZETA*DTDZ(N-1)-DTDZ(N-2)+(1.-ZETA)*DTDZ(N))/EPS
DCDZ(N,1)=(ZETA*CS(N-1)-CS(N-2)+(1.-ZETA)*CS(N))/EPS
DDHDZ(N)=(ZETA*D(N-1,1)-D(N-2,1)+(1.-ZETA)*D(N,1))/EPS
DDHDZ(1)=(D(1,1)*(1.-EP1SQ)+D(2,1)*EP1SQ-D(3,1))/SIGMA

C
C
C   COMPUTE A1,B1,C1, AND G VECTORS

DO 80 I=1,N
UNPSI=1.+DSDT(I)*10800./(CS(I)*CSUBV)
ANEW(I)=SHR*D(I,1)/UNPSI
BNEW(I,1)=DDHDZ(I)*SHR/UNPSI+DCDZ(I,1)*SHR*D(I,1)/(UNPSI*CS(I))-WA
1P(I)
CNEW(I,1)=DCDZ(I,1)*R*(DZDT+WAP(I))/(CS(I)*CSUBV*UNPSI)
RSUBR=0.

C
C#####
C
C   IF (G(I,3).LT.0.) RSUBR=G(I,3)

C
C#####
G(I,1)=10800.*RSUBR/(CS(I)*CSUBV)-SHR*(D(I,1)*DDTDZ(I)+D(I,1)*DTDZ
1(I)*DCDZ(I,1)/CS(I)+DTDZ(I)*DDHDZ(I))
80 G(I,1)=G(I,1)/UNPSI
IF (IFLIP.EQ.0) GO TO 100
DO 90 I=1,N
A1(I,1)=(A1(I,1)+ANEW(I))/2.
B1(I)=(B1(I)+BNEW(I,1))/2.
90 C1(I)=(C1(I)+CNEW(I,1))/2.
GO TO 120
100 DO 110 I=1,N
A1(I,1)=ANEW(I)
B1(I)=BNEW(I,1)
110 C1(I)=CNEW(I,1)
IFLIP=1
120 RETURN
END

```

TABLE 30

Listings of the Class III Functions SAT and DSAT

```

C
C
C
C
FUNCTION SAT (TK)
  DIMENSION A(7)
  DATA A/6.107799961,4.436518521E-1,1.428945805E-2,2.650648471E-4,3.
1031240396E-6,2.034080948E-8,6.136820929E-11/

  ROUTINE FOR CALCULATING THE SATURATION CONCENTRATION OF
  WATER VAPOR (GRAM-MOLES/CC) AT TK DEGREES KELVIN

  TC=TK-273.15
  ESAT=A(1)+TC*(A(2)+TC*(A(3)+TC*(A(4)+TC*(A(5)+TC*(A(6)+TC*A(7))))))
1)
  SAT=ESAT*1.2033E-5/TK
  RETURN
  END

FUNCTION DSAT (TK)
  DIMENSION A(7)
  DATA A/6.107799961,4.436518521E-1,1.428945805E-2,2.650648471E-4,3.
1031240396E-6,2.034080948E-8,6.136820929E-11/

  ROUTINE FOR CALCULATING THE TEMPERATURE DERIVATIVE-
  OF THE SATURATION CONCENTRATION OF WATER VAPOR,
  C
  C
  C
  OUTPUT UNITS IN MOLES/(CC*DEGK).

  TC=TK-273.2
  DSAT=A(2)+TC*(2.*A(3)+TC*(3.*A(4)+TC*(4.*A(5)+TC*(5.*A(6)+TC*(6.*A
1(7))))))
  DSAT=1.2033E-5*DSAT/TK-SAT(TK)/TK
10
  RETURN
  END

```


Class III subroutine HENRY is presented in Table 31. This subroutine was borrowed directly from a previous code [Drewes and Hales (1977)], and computes a solubility coefficient HPRIME for SO_2 in water, which is defined by the equation

$$y = \text{HPRIME } c$$

Here y denotes the mole fraction of SO_2 in the gas phase, and c is its equilibrium concentration (moles/cm³) in water. Arguments in the call sequence are as follows:

- c mole fraction SO_2 in gas phase *or* concentration SO_2 (moles total dissolved SO_2/cm^3) in water
- HEX acidity of water in moles/liter
- T temperature, °K
- P pressure, atmospheres
- LGOPT flag denoting whether c is supplied as a gas-phase or aqueous-phase value.

I. Deck Structure and Input Data for Example Application 2

The requirements for execution of Example 2 essentially parallel those described for Example 1. These are itemized sequentially in the following paragraphs.

- The main program must be available. This is identical to the main program used for Example 1, and is listed in Appendix B.
- The Class II subroutines must be available. These also are identical to those used for the previous example, and are listed in Appendix B.
- The user-modified Class I subroutines must be available; appropriate versions are:

GRDSET	Table 3
VERT	Table 22
TIMESP	Table 23
BOUND	Table 24

TABLE 31

Listing of the Class III Subroutine HENRY

```

SUBROUTINE HENRY (HPRIME,C,HEX,T,P,LGOPT)
C .....SUBROUTINE FOR CALCULATING APPARANT HENRY'S LAW CONSTANT U
C ..... JOHNSTONE-LEPPLA PARAMETERS
C ..... HPRIME HAS UNITS OF CM**3/MOLE
C ..... C IS M. F. (GAS) OR MOLE/CM**3 (LIQUID)
C ..... HEX IS MOLAR
C ..... T IS DEGREES K
C ..... P IS ATMOSPHERES
C ..... LGOPT IS A FLAG: 3HGAS OR 3HLIQ
HANK=EXP(10.21-3112./T)/P
EQCON=EXP(-10.96+1970./T)
B=EQCON+HEX
IF (LGOPT.EQ.3HGAS) GO TO 20
IF (LGOPT.NE.3HLIQ) WRITE (6,50)
LGOPT=3HLIQ
CDUM=EQCON*1000.*C
C .....TEST FOR LOW CONCENTRATION CONDITIONS
TEST1=4.*CDUM/R/B
IF (TEST1.GT..001) GO TO 10
C .....ASYMPTOTIC EXPRESSION FOR LOW CONCENTRATION CONDITIONS
HPRIME=HANK*(1.-EQCON/B)
GO TO 40
C .....SOLUTION OF TOTAL SOLUBILITY EQUATION
10 HPRIME=HANK*(1.+(B-SQRT(B*B+4.*CDUM))/2000./C)
GO TO 40
C .....TEST FOR LOW CONCENTRATION CONDITIONS
20 TEST1=4.*EQCON*C/HEX/HEX/HANK
IF (TEST1.GT..001) GO TO 30
C .....ASYMPTOTIC EXPRESSION FOR LOW CONCENTRATION CONDITIONS
HPRIME=HANK*HEX/B
GO TO 40
C .....SOLUTION OF TOTAL SOLUBILITY EQUATION
30 HPRIME=C/(C/HANK-(HEX-SQRT(HEX*HEX+4.*C*EQCON/HANK))/2.)
40 HPRIME=HPRIME*1000.
RETURN
50 FORMAT (48H WARNING---LGOPT IN HPRIME HAS IMPROPER VALUE---,16H"LI
1QUID" ASSUMED)
END

```

CONC	Table 25
DIFF	Table 26
SETTL	Table 27
GEN	Table 28
PRESSF	Table 12
TEMPF	Table 29

- The Class III subroutines SAT, DSAT, and HENRY (Tables 30 and 31) must be available.
- The input data must be available. Table 32 gives the formatted input data set appropriate to this case example.

J. Example Output for Example Application 2

Selected output for Example 2 is given in Tables 33-40. Tables 33-35 describe the stipulated initial conditions (cf. Table 21) for computed, constrained, and selected atmospheric variables, respectively, corresponding to the 0.1 micron sulfate particle case. As noted in this previous text all units are given in °K, newtons, gram-moles, and centimeters. As can be observed from Table 34 the model cloud base begins at 1950 meters above the datum at a temperature of 269°K; this corresponds to a supercooled liquid cloud. The cloud-borne primary sulfate is set at a constant level of $.157\text{E-}14$ moles/cm³; this results from the nucleation parameterization scheme, which allows only a set maximum number of particles (chosen to be 300 here) to be active nuclei in a given cubic centimeter of air at any given time (cf. Table 25, subroutine CONC, where XNC is set equal to 300). It is important also to note that the aqueous-phase concentrations given in Tables 33-41 denote moles per cubic centimeter *total space*, rather than moles per cubic centimeter water.

Tables 36-38 provide corresponding results for $.1177\text{E}+05$ seconds, or about 3.27 hours of simulation time. As can be observed from Figure 5, this corresponds to an ascending air column as it reaches the top of the incline. Precipitation is reaching the ground, and the cloud base is now about one thousand meters above datum. Comparison of the GCS04P columns in Tables 33 and 36 indicates that sulfate has been depleted significantly from the atmosphere by the scavenging process, although not so much as to alter the $.157\text{E-}14$ cloud-borne

TABLE 32

Input Data for Example 2*

	8	4				
TEMP						
V+CH2O						
PRECIP						
RS04S						
RS04P						
CS04S						
GS02						
GCS04P						
CS04P						
CLOUD						
CS02						
GOZONE						
	21	10	3	1.50E4	2.40E4	3.00E2
	1	1	1	1	1	1
	1	1	1			
	1	1	2			
	1	0	3			
	1	0	3			
	1	0	3			
	1	1	2			
	1	1	2			
	1	1	2			

* In above species list the following naming convention is used:
V = vapor; C = cloud; R = rain; G = gas; P = primary; S = secondary.

TABLE 33

Initial Computed Variables for Example Application 2
0.1 Micron Aerosol Particles

COMPUTED RESULTS FOR 0.

SECONDS OF MODEL TIME, PAGE 1

CURRENT TIME INCREMENT IS .3000E+03SECONDS

LOWER BOUNDARY POSITION ABOVE DATUM = 0.

CURRENT ASCENT RATE = 0.

HEIGHT	TEMP	V+CH2O	PRECIP	RS04S	RS04P	CS04S	GS02	GCS04P
.3000E+06	.262E+03	.154E-06	0.	0.	0.	0.	.392E-12	.392E-13
.2850E+06	.263E+03	.161E-06	0.	0.	0.	0.	.397E-12	.397E-13
.2700E+06	.264E+03	.168E-06	0.	0.	0.	0.	.403E-12	.403E-13
.2550E+06	.265E+03	.175E-06	0.	0.	0.	0.	.409E-12	.409E-13
.2400E+06	.266E+03	.183E-06	0.	0.	0.	0.	.415E-12	.415E-13
.2250E+06	.267E+03	.191E-06	0.	0.	0.	0.	.421E-12	.421E-13
.2100E+06	.268E+03	.200E-06	0.	0.	0.	0.	.427E-12	.427E-13
.1950E+06	.269E+03	.209E-06	0.	0.	0.	0.	.432E-12	.432E-13
.1800E+06	.271E+03	.218E-06	0.	0.	0.	0.	.438E-12	.438E-13
.1650E+06	.272E+03	.228E-06	0.	0.	0.	0.	.443E-12	.443E-13
.1500E+06	.273E+03	.238E-06	0.	0.	0.	0.	.449E-12	.449E-13
.1350E+06	.275E+03	.249E-06	0.	0.	0.	0.	.455E-12	.455E-13
.1200E+06	.276E+03	.260E-06	0.	0.	0.	0.	.460E-12	.460E-13
.1050E+06	.278E+03	.272E-06	0.	0.	0.	0.	.466E-12	.466E-13
.9000E+05	.279E+03	.284E-06	0.	0.	0.	0.	.472E-12	.472E-13
.7500E+05	.281E+03	.297E-06	0.	0.	0.	0.	.478E-12	.478E-13
.6000E+05	.282E+03	.310E-06	0.	0.	0.	0.	.484E-12	.484E-13
.4500E+05	.284E+03	.324E-06	0.	0.	0.	0.	.491E-12	.491E-13
.3000E+05	.285E+03	.339E-06	0.	0.	0.	0.	.497E-12	.497E-13
.1500E+05	.287E+03	.354E-06	0.	0.	0.	0.	.503E-12	.503E-13
0.	.288E+03	.370E-06	0.	0.	0.	0.	.510E-12	.510E-13

TABLE 34

Constrained Variables for Example Application 2
0.1 Micron Aerosol Particles

CONSTRAINED RESULTS FOR 0. SECONDS OF MODEL TIME, PAGE 1
CURRENT TIME INCREMENT IS .3000E+03SECONDS

LOWER BOUNDARY POSITION ABOVE DATUM = 0. CURRENT ASCENT RATE = 0.

HEIGHT	CSO4P	CLOUD	CSO2	GOZONE
.3000E+06	.157E-14	.358E-07	.120E-14	.169E-11
.2850E+06	.157E-14	.332E-07	.108E-14	.171E-11
.2700E+06	.157E-14	.304E-07	.957E-15	.174E-11
.2550E+06	.157E-14	.275E-07	.839E-15	.176E-11
.2400E+06	.157E-14	.244E-07	.722E-15	.179E-11
.2250E+06	.157E-14	.211E-07	.607E-15	.181E-11
.2100E+06	.157E-14	.177E-07	.495E-15	.184E-11
.1950E+06	.157E-14	.645E-08	.172E-15	.186E-11
.1800E+06	0.	0.	0.	.189E-11
.1650E+06	0.	0.	0.	.191E-11
.1500E+06	0.	0.	0.	.194E-11
.1350E+06	0.	0.	0.	.196E-11
.1200E+06	0.	0.	0.	.199E-11
.1050E+06	0.	0.	0.	.201E-11
.9000E+05	0.	0.	0.	.204E-11
.7500E+05	0.	0.	0.	.206E-11
.6000E+05	0.	0.	0.	.209E-11
.4500E+05	0.	0.	0.	.212E-11
.3000E+05	0.	0.	0.	.214E-11
.1500E+05	0.	0.	0.	.217E-11
0.	0.	0.	0.	.220E-11

TABLE 35

Initial Atmospheric Pressures and Air Concentrations

PRESSURE PROFILE

HEIGHT	PRESSURE	AIR CONC
.3000E+06	.7071E+01	.3250E-04
.2850E+06	.7200E+01	.3296E-04
.2700E+06	.7330E+01	.3343E-04
.2550E+06	.7463E+01	.3391E-04
.2400E+06	.7599E+01	.3440E-04
.2250E+06	.7737E+01	.3490E-04
.2100E+06	.7877E+01	.3541E-04
.1950E+06	.8020E+01	.3585E-04
.1800E+06	.8166E+01	.3631E-04
.1650E+06	.8314E+01	.3677E-04
.1500E+06	.8465E+01	.3723E-04
.1350E+06	.8618E+01	.3771E-04
.1200E+06	.8775E+01	.3819E-04
.1050E+06	.8934E+01	.3867E-04
.9000E+05	.9096E+01	.3917E-04
.7500E+05	.9261E+01	.3967E-04
.6000E+05	.9429E+01	.4018E-04
.4500E+05	.9600E+01	.4070E-04
.3000E+05	.9775E+01	.4123E-04
.1500E+05	.9952E+01	.4176E-04
0.	.1013E+02	.4230E-04

TABLE 36

Computed Variables for Example Application 2
0.1 Micron Aerosol Particles

COMPUTED RESULTS FOR .1177E+05 SECONDS OF MODEL TIME, PAGE 1

CURRENT TIME INCREMENT IS .1081E+03SECONDS

LOWER BOUNDARY POSITION ABOVE DATUM = .5818E+05 CURRENT ASCENT RATE = .1000E+02

HEIGHT	TEMP	V+CH2O	PRECIP	RS04S	RS04P	CS04S	GS02	GCS04P
.3582E+06	.257E+03	.103E-06	.233E-08	.201E-15	.234E-15	.164E-14	.357E-12	.294E-13
.3432E+06	.259E+03	.105E-06	.505E-08	.438E-15	.579E-15	.121E-14	.363E-12	.295E-13
.3282E+06	.260E+03	.110E-06	.684E-08	.561E-15	.898E-15	.812E-15	.370E-12	.293E-13
.3132E+06	.261E+03	.117E-06	.824E-08	.626E-15	.120E-14	.567E-15	.377E-12	.291E-13
.2982E+06	.262E+03	.126E-06	.952E-08	.669E-15	.151E-14	.430E-15	.383E-12	.291E-13
.2832E+06	.263E+03	.136E-06	.105E-07	.688E-15	.179E-14	.352E-15	.389E-12	.293E-13
.2682E+06	.264E+03	.147E-06	.111E-07	.674E-15	.197E-14	.300E-15	.395E-12	.299E-13
.2532E+06	.265E+03	.159E-06	.124E-07	.710E-15	.231E-14	.261E-15	.401E-12	.306E-13
.2382E+06	.266E+03	.172E-06	.142E-07	.767E-15	.273E-14	.236E-15	.406E-12	.317E-13
.2232E+06	.267E+03	.186E-06	.139E-07	.707E-15	.274E-14	.212E-15	.412E-12	.328E-13
.2082E+06	.269E+03	.201E-06	.148E-07	.717E-15	.301E-14	.187E-15	.418E-12	.339E-13
.1932E+06	.270E+03	.217E-06	.159E-07	.734E-15	.331E-14	.169E-15	.423E-12	.351E-13
.1782E+06	.271E+03	.233E-06	.158E-07	.684E-15	.333E-14	.145E-15	.429E-12	.365E-13
.1632E+06	.272E+03	.250E-06	.177E-07	.725E-15	.379E-14	.124E-15	.435E-12	.382E-13
.1482E+06	.273E+03	.267E-06	.185E-07	.766E-15	.420E-14	.978E-16	.441E-12	.402E-13
.1332E+06	.274E+03	.284E-06	.184E-07	.715E-15	.426E-14	.679E-16	.447E-12	.423E-13
.1182E+06	.275E+03	.300E-06	.208E-07	.623E-15	.461E-14	.316E-16	.453E-12	.442E-13
.1032E+06	.275E+03	.316E-06	.193E-07	.761E-15	.526E-14	.244E-17	.459E-12	.461E-13
.8818E+05	.277E+03	.330E-06	.166E-07	.866E-15	.521E-14	0.	.466E-12	.471E-13
.7318E+05	.278E+03	.341E-06	.172E-07	.738E-15	.506E-14	0.	.474E-12	.478E-13
.5818E+05	.279E+03	.349E-06	.179E-07	.636E-15	.558E-14	0.	.482E-12	.486E-13

TABLE 37

Constrained Variables for Example Application 2
0.1 Micron Aerosol Particles

CONSTRAINED RESULTS FOR .1177E+05 SECONDS OF MODEL TIME, PAGE 1
CURRENT TIME INCREMENT IS .1081E+03SECONDS

LOWER BOUNDARY POSITION ABOVE DATUM = .5818E+05 CURRENT ASCENT RATE = .1000E+02

HEIGHT	CSO4P	CLOUD	CSO2	GOZONE
.3582E+06	.157E-14	.188E-07	.672E-15	.169E-11
.3432E+06	.157E-14	.133E-07	.453E-15	.171E-11
.3282E+06	.157E-14	.102E-07	.332E-15	.174E-11
.3132E+06	.157E-14	.859E-08	.269E-15	.176E-11
.2982E+06	.157E-14	.780E-08	.235E-15	.179E-11
.2832E+06	.157E-14	.739E-08	.214E-15	.181E-11
.2682E+06	.157E-14	.711E-08	.198E-15	.184E-11
.2532E+06	.157E-14	.685E-08	.183E-15	.186E-11
.2382E+06	.157E-14	.679E-08	.174E-15	.189E-11
.2232E+06	.157E-14	.672E-08	.166E-15	.191E-11
.2082E+06	.157E-14	.649E-08	.154E-15	.194E-11
.1932E+06	.157E-14	.647E-08	.148E-15	.196E-11
.1782E+06	.157E-14	.622E-08	.136E-15	.199E-11
.1632E+06	.157E-14	.605E-08	.128E-15	.201E-11
.1482E+06	.157E-14	.578E-08	.118E-15	.204E-11
.1332E+06	.157E-14	.524E-08	.102E-15	.206E-11
.1182E+06	.157E-14	.416E-08	.763E-16	.209E-11
.1032E+06	.157E-14	.196E-08	.287E-16	.212E-11
.8818E+05	0.	0.	0.	.214E-11
.7318E+05	0.	0.	0.	.217E-11
.5818E+05	0.	0.	0.	.220E-11

TABLE 38

Atmospheric Pressures and Air Concentrations
at .1177E+05 Simulation Time

PRESSURE PROFILE

HEIGHT	PRESSURE	AIR CONC
.3582E+06	.6605E+01	.3086E-04
.3432E+06	.6725E+01	.3129E-04
.3282E+06	.6847E+01	.3173E-04
.3132E+06	.6971E+01	.3217E-04
.2982E+06	.7098E+01	.3262E-04
.2832E+06	.7227E+01	.3307E-04
.2682E+06	.7358E+01	.3353E-04
.2532E+06	.7491E+01	.3399E-04
.2382E+06	.7627E+01	.3446E-04
.2232E+06	.7766E+01	.3493E-04
.2082E+06	.7907E+01	.3542E-04
.1932E+06	.8050E+01	.3591E-04
.1782E+06	.8196E+01	.3642E-04
.1632E+06	.8345E+01	.3694E-04
.1482E+06	.8496E+01	.3748E-04
.1332E+06	.8651E+01	.3803E-04
.1182E+06	.8808E+01	.3859E-04
.1032E+06	.8967E+01	.3917E-04
.8818E+05	.9130E+01	.3970E-04
.7318E+05	.9296E+01	.4022E-04
.5818E+05	.9465E+01	.4075E-04

TABLE 39

Computed Variables for Example Application 2
0.3 Micron Aerosol Particles

COMPUTED RESULTS FOR .1177E+05 SECONDS OF MODEL TIME, PAGE 1

CURRENT TIME INCREMENT IS .1081E+03SECONDS

LOWER BOUNDARY POSITION ABOVE DATUM = .5818E+05 CURRENT ASCENT RATE = .1000E+02

HEIGHT	TEMP	V+CH20	PRECIP	RS04S	RS04P	CS04S	GS02	GCS04P
.3582E+06	.257E+03	.103E-06	.233E-08	.204E-15	.766E-16	.170E-14	.358E-12	.382E-15
.3432E+06	.259E+03	.105E-06	.505E-08	.444E-15	.184E-15	.128E-14	.365E-12	.296E-15
.3282E+06	.260E+03	.110E-06	.684E-08	.572E-15	.267E-15	.887E-15	.371E-12	.195E-15
.3132E+06	.261E+03	.117E-06	.824E-08	.642E-15	.340E-15	.647E-15	.378E-12	.130E-15
.2982E+06	.262E+03	.126E-06	.952E-08	.689E-15	.416E-15	.510E-15	.384E-12	.101E-15
.2832E+06	.263E+03	.136E-06	.105E-07	.710E-15	.492E-15	.426E-15	.390E-12	.101E-15
.2682E+06	.264E+03	.147E-06	.111E-07	.693E-15	.564E-15	.364E-15	.396E-12	.120E-15
.2532E+06	.265E+03	.159E-06	.124E-07	.725E-15	.709E-15	.313E-15	.401E-12	.152E-15
.2382E+06	.266E+03	.172E-06	.142E-07	.774E-15	.905E-15	.278E-15	.407E-12	.200E-15
.2232E+06	.267E+03	.186E-06	.139E-07	.701E-15	.963E-15	.243E-15	.413E-12	.265E-15
.2082E+06	.269E+03	.201E-06	.148E-07	.692E-15	.129E-14	.205E-15	.418E-12	.368E-15
.1932E+06	.270E+03	.217E-06	.159E-07	.687E-15	.177E-14	.172E-15	.424E-12	.597E-15
.1782E+06	.271E+03	.233E-06	.158E-07	.621E-15	.163E-14	.127E-15	.429E-12	.107E-14
.1632E+06	.272E+03	.250E-06	.177E-07	.629E-15	.234E-14	.831E-16	.435E-12	.215E-14
.1482E+06	.273E+03	.267E-06	.185E-07	.603E-15	.567E-14	.417E-16	.441E-12	.448E-14
.1332E+06	.274E+03	.284E-06	.184E-07	.532E-15	.695E-14	.143E-16	.447E-12	.939E-14
.1182E+06	.275E+03	.300E-06	.208E-07	.490E-15	.120E-13	.287E-17	.453E-12	.188E-13
.1032E+06	.275E+03	.316E-06	.193E-07	.489E-15	.281E-13	.251E-18	.459E-12	.399E-13
.8818E+05	.277E+03	.330E-06	.166E-07	.478E-15	.303E-13	0.	.466E-12	.504E-13
.7318E+05	.278E+03	.341E-06	.172E-07	.440E-15	.253E-13	0.	.474E-12	.507E-13
.5818E+05	.279E+03	.349E-06	.179E-07	.430E-15	.228E-13	0.	.482E-12	.512E-13

TABLE 40

Constrained Variables for Example Application 2
0.3 Micron Aerosol Particles

CONSTRAINED RESULTS FOR .1177E+05 SECONDS OF MODEL TIME, PAGE 1

CURRENT TIME INCREMENT IS .1081E+03SECONDS

LOWER BOUNDARY POSITION ABOVE DATUM = .5818E+05 CURRENT ASCENT RATE = .1000E+02

HEIGHT	CS04P	CLOUD	CS02	GOZONE
.3582E+06	.382E-15	.188E-07	.687E-15	.169E-11
.3432E+06	.296E-15	.133E-07	.468E-15	.171E-11
.3282E+06	.195E-15	.102E-07	.347E-15	.174E-11
.3132E+06	.130E-15	.859E-08	.285E-15	.176E-11
.2982E+06	.101E-15	.780E-08	.251E-15	.179E-11
.2832E+06	.101E-15	.739E-08	.230E-15	.181E-11
.2682E+06	.120E-15	.711E-08	.213E-15	.184E-11
.2532E+06	.152E-15	.685E-08	.198E-15	.186E-11
.2382E+06	.200E-15	.679E-08	.189E-15	.189E-11
.2232E+06	.265E-15	.672E-08	.179E-15	.191E-11
.2082E+06	.368E-15	.649E-08	.166E-15	.194E-11
.1932E+06	.597E-15	.647E-08	.157E-15	.196E-11
.1782E+06	.107E-14	.622E-08	.140E-15	.199E-11
.1632E+06	.215E-14	.605E-08	.120E-15	.201E-11
.1482E+06	.448E-14	.578E-08	.897E-16	.204E-11
.1332E+06	.939E-14	.524E-08	.501E-16	.206E-11
.1182E+06	.188E-13	.416E-08	.184E-16	.209E-11
.1032E+06	.399E-13	.196E-08	.225E-17	.212E-11
.8818E+05	0.	0.	0.	.214E-11
.7318E+05	0.	0.	0.	.217E-11
.5818E+05	0.	0.	0.	.220E-11

concentrations. It is interesting also to observe the wavy character of the vertical profile of precipitation. This indicates a showery behavior, which will be discussed in more detail shortly.

Tables 39 and 40 present calculations at this same time step for a stipulated sulfate effective particle size of 0.3 microns. In contrast to Tables 40 and 41 these indicate variable cloud-borne primarily sulfate concentrations, which arise because the sulfate particles are generally less abundant than the critical 300 particle per cubic centimeter level. This is reflected by much higher rainborne primary sulfate concentrations and scavenging rates. Reactive scavenging of SO_2 is suppressed in this 0.3-micron situation, owing to the greater inhibiting effect of sulfate-induced hydrogen ion.

More general aspects of the model simulation are presented in Figures 5-8. As indicated in Figure 5 the condensing cloud, initially containing no rain, begins to produce virga prior to the air column's ascent. The resulting downward flux of precipitation erodes the cloud, but this is more than compensated by cloud buildup as ascent and expansion occur.

The increased cloud buildup results in more precipitation, which reaches the surface as a light mist near the top of the incline (at about 2.8 hours), with moderate rainfall occurring immediately thereafter. The depletion of atmospheric water resulting from this rainfall erodes the cloud rapidly after the plateau is reached; after descent on the backslope, this depletion plus warming by compression completely evaporates the remaining cloud system. This latter combination creates a mild chinook (foehn) situation; which can be observed in the computed temperature profiles.

As noted above the rainfall pattern generated in this example application is decidedly showery in nature. This is illustrated by Figure 6, which indicates a number of rainbands, or showers, to emerge early in the storm's history. Such pronounced behavior is felt to be a somewhat unrealistic portrayal of actual storms of this type, although some interesting insights regarding precipitation behavior emerge as a result. Analysis of the detailed model output shows the showery behavior to originate from nonlinear properties in the rain generation-removal process. Referring again to Table 20, one notes

FIGURE 6: Computed Precipitation Rates for Model Storm

FIGURE 7: Computed Primary (solid) and Secondary (dashed) Sulfate Concentrations in Ground-Level Rainwater; 0.1-micron Aerosol Case

FIGURE 8: Computed Primary (solid) and Secondary (dashed) Sulfate Concentrations in Ground-Level Rainwater; 0.3-micron Aerosol Case

that rain-production steps W2 and W3 are basically nonlinear, and constitute something of a positive-feedback system. Removal steps W4 and W5 are also nonlinear, and this situation corresponds roughly to the filling of a bucket, with repeated dumps at various intervals of time.

Certainly a portion of this showery behavior could be modified by revised choices of the parameterizations summarized in Table 20; in addition, these fluctuations could be attenuated appreciably if a more gradual ascent rate were chosen. Moreover, much of the storm's showery nature is expected to result from the model's one-dimensional nature. There are two reasons for this. Firstly, vertical wind shears, if present, would almost certainly dampen this variability by horizontal mixing of adjacent regions. Such behavior is not treated by the model. Secondly, horizontal divergences occurring during ascent of real air parcels often can be expected to reduce moisture concentrations below those computed on the basis of one-dimensionality. High moisture concentrations generally accentuate the positive-feedback situation described above, and are thus more conducive to showery situations.

Concentrations of primary and secondary sulfate in surface rainwater, as computed by the model, are shown in Figures 7 and 8. Figure 7 corresponds to the assumption of a homogeneous 0.1-micron primary sulfate aerosol; Figure 8 is derived from the assumption of a 0.3-micron aerosol, having the same initial molar concentration.

Addressing first the primary sulfate curve in Figure 7, it is observed that high concentrations occur initially, with a rapid fall-off followed by a partial recovery at the storm's end. The large initial concentrations result primarily from evaporation effects in the virga, immediately preceeding contact with the ground.

The first portion of the initial high-concentration region is associated with the pre-shower mist described earlier. The second portion occurs during the onset of the first significant shower period. The rapid concentration decrease after about 3.1 hours of model time arises primarily from diminishing importance of the evaporation effect, which reemerges as a significant factor during final stages of the storm.

In concordance with Table 36 the secondary sulfate curve in Figure 7 suggests that reaction (32) should have a less important, but significant, contributory effect under the circumstances depicted by the model. Similar shapes between the two curves demonstrate further the importance of evaporation effects in concentrating pollutant in rainwater as it falls below the cloud system.

The 0.3-micron aerosol curves in Figure 8 exhibit some features which differ substantially from their 0.1-micron counterparts. Extremely high initial concentrations of primary sulfate are produced early in the storm's history, both in the pre-shower mist and early in the first shower. Following this the concentrations drop off rapidly, although the computed concentrations remain higher than those of the 0.1 micron computations throughout most of the storm.

The chief reason for the difference in initial scavenging rates between the 0.1 and 0.3 micron aerosol cases is, of course, based upon the nucleation mechanism assumed in this particular application of the model. A cloud droplet of given size nucleating upon a 0.3-micron particle will experience twenty-seven times the sulfate concentration of one nucleating on a 0.1 micron particle, and this effect influences profoundly the behavior of the overall system. For the 0.3-micron case this enhanced scavenging rate is so great that, by the end of the storm, much of the available primary sulfate has been removed from the system. As a consequence, final primary-sulfate concentrations in Figure 7 are beginning to overtake those in Figure 8, and would actually exceed them had the storm been longer in duration.

As noted previously in the context of Table 39, scavenging of secondary sulfate is suppressed early in the storm for the 0.3-micron case, because of the increased primary sulfate concentrations and the resulting inhibition of acidity on reaction (32). During later stages of the storm, when primary-sulfate concentrations are roughly equal between the two cases, the secondary sulfate curves correspond almost exactly.

Execution times for each of the two computer runs in this example (0.1 and 0.3 micron aerosol cases) were 15.5 seconds on the CDC 6400. This corresponds to 24000 seconds (or 6.67 hours) of model time.

VI. PROBLEMS WITH PLUVIUS: SOME PRACTICAL DIFFICULTIES AND SUGGESTIONS FOR THEIR RESOLUTION

Aside from the inherent one-dimensionality of PLUVIUS, there are a number of practical difficulties which can emerge with the model to produce malfunctions or inappropriate results, and several actions can be taken to minimize these effects. The first and most important of these is simply the problem of formulating a conceptual model and implementing it on PLUVIUS in a correct and expedient fashion. There is no easy solution to this; but our experience has shown that significant shakedown and backtracking time can be eliminated via systematic use of a formal implementation procedure. The procedure is as follows:

- Create interaction diagrams (eg. Figures 3 and 4) to define all pathways and dependent variables
- Create a conversion table (eg. Table 20) to define quantitatively the interconversion parameterizations to be employed
- Create a table of initial conditions (eg. Table 21)
- Determine the number of constrained and computed variables, and the number of diffusion regimes
- Adjust the code's dimensioning to be sufficiently large to accommodate these parameters (cf. Section IIA)
- Ensure that the water-related variables (if employed) are ordered in a manner that is consistent with usage in the main program and subroutine TEMPF (denoted by ##### ...)
- Define all required boundary conditions
- Define grid height and temporal and spatial spacing
- Define vertical movement of grid and relative vertical flows (w_A')
- Define fall-velocity features
- Define diffusional behavior

- Revise the Class I subroutines to conform with the above specifications. A check-off list for this purpose is as follows:
 - 1) GRDSET
 - 2) TIMESP
 - 3) VERT
 - 4) BOUND
 - 5) CONC
 - 6) DIFF
 - 7) SETTLE
 - 8) GEN
 - 9) TEMPF
 - 10) PRESSF
- Consolidate the above subroutines with PLUVIUS and the Class II subroutines PRINT, TRIDAG, SETVEC and DERIV
- Add any required Class III subroutines
- Create an input data set as described in Section IIIB (eg. Table 15)
- Execute the program.

A second problem with PLUVIUS, which may arise to plague the user, is that of stiff or rapid interconversion processes. These always can be countered by reducing temporal grid spacing; however, the associated increase in computation cost may become intolerable. *Stiff* systems can be accommodated in a somewhat more efficient manner by redefining problem variables and expressing their *effective* conversion rates in subroutine GEN. The effective conversion rates can be generated using an efficient "zero-dimensional" integrator, such as the Gear (1971) routine. Both rapid and stiff systems can be countered successfully using a more elegant splitting scheme, such as the one described by Easter, et al (1981). This last procedure, however, would require considerable restructuring of the PLUVIUS code.

Numerical diffusion and related problems can pose rather severe computational difficulties if advection terms in the z direction become comparatively large, especially if they accompany substantial z gradients of concentration. These

effects were readily apparent in the results of Example 2, particularly for rainborne sulfate in virga. Under these conditions the rainborne sulfate conformed essentially with a downward-moving step function, and the Crank-Nicholson procedure's difficulty in dealing with this was manifest in some rather noisy concentration profiles, which even had a tendency to go negative on occasion. Unfortunately not much can be done within PLUVIUS to counter this situation, short of radically reducing temporal and spatial grid spacing. Noting that discontinuities such as those encountered here do not often occur in nature, however, suggests that if one were to employ a more physically-satisfactory manner of characterizing the process, much of this problem could be eliminated. Example 2's presumption that all rainborne sulfate is retained by the rain until the last drop evaporates, whereupon it is suddenly and totally released, for example, could certainly be modified to produce a more elegant and less numerically troublesome treatment.

Returning to the question of one-dimensionality it should be noted that although this is a major shortcoming of PLUVIUS, its effects can be minimized by appropriate applications within the code. Three-dimensional plumes, for example, may be simulated by diluting the air parcel to account for cross-plume mixing by providing time-dependent dilution factors within subroutine CONC or subroutine GEN, depending upon whether the user desires to simulate this process in an absolute or differential manner. Three-dimensional flow fields can be approximated in a similar manner, with judicious choice of the vertical-velocity component w_A' .

In a final analysis it is once again noted that PLUVIUS embodies a somewhat gross approximation to the real world, and constitutes a rather imperfect "lens" through which to observe atmospheric behavior. With this qualification, however, it is expected that prudent application of the code can lead to a number of revealing and scientifically rewarding insights. It is with this intent that PLUVIUS is submitted to the user community.

REFERENCES

- Berry, E. X. and Reinhardt, R. L., (1974). An Analysis of Cloud Drop Growth by Collection. J. Atmos. Sci., 31, 1814-2127.
- Bird, R. B., Stewart, W. E. and Lightfoot, E. N., (1960). Transport Phenomena, Wiley, New York.
- Drewes, D. R. and Hales, J. M., (1977). SMICK - A Scavenging Model Incorporating Chemical Kinetics. Final Report on RP-858, Electric Power Research Institute, Palo Alto, California.
- Easter, R. C., Hales, J. M., Sverdrup, G. M. and Spicer, C. W., (1981). Plume Conversion Rates in the SURE Region. Final Report on RP-860, Volume 3. Electric Power Research Institute, Palo Alto, California.
- Gear, C. W., (1971). The Automatic Integration of Ordinary Differential Equations. Comm. ACM, 14, 176-190.
- Hales, J. M. and Sutter, S., (1973). Solubility of Sulfur Dioxide in Water at Low Concentrations. Atm. Envi., 7, 977-1001.
- Heines, T. S. and Peters, L. K., (1973). The Effect of a Horizontal Impervious Layer Caused by a Temperature Inversion Aloft on the Dispersion of Pollutants in the Atmosphere. Atm. Envi., 7, 39-48.
- Kessler, E., (1969). On The Distribution and Continuity of Water Substance in Atmospheric Circulations. AMS Meteorological Monograph, Vol. 10, Am. Met. Soc., Boston, Massachusetts.
- Kirk, R. E. and Othmer, D. F., (1947). Encyclopedia of Chemical Technology. Interscience Encyclopedia, New York.
- Mason, B. J., (1957). The Physics of Clouds. Oxford University Press, Oxford.
- Omstedt, G. and Rodhe, H., (1978). Transformation and Removal Processes for Sulfur Compounds as Described on a One-Dimensional, Time-Dependent Diffusion Model. Atm. Envi., 12, 503-509.
- Penkett, S. A., Jones, B. M. R., Brice, K. A. and Eggleton, A. E. J., (1979). The Importance of Atmospheric Ozone and Hydrogen Peroxide in Oxidising Sulphur Dioxide in Cloud and Rainwater. Atm. Envi., 13, 123-137.
- Pruppacher, H. R. and Klett, J. D., (1978). Microphysics of Clouds and Precipitation. Reidel, Boston, Massachusetts.
- Scott, B. C., (1981). Predictions of In-Cloud Conversion Rates of SO₂ to SO₄ Based Upon a Simple Chemical and Kinematic Storm Model. Atm. Envi., in press.

APPENDIX A

FINITE-DIFFERENCE APPROXIMATION

FINITE DIFFERENCE APPROXIMATION

The finite difference procedure employed in this code is a variable-grid extension of the basic Crank-Nicholson method (Carnahan, Luther, and Wilkes (1969)). It is similar in most respects to the differencing scheme described by Sundqvist and Veronis (1979); it is characterized by the features of unconditional stability, and a truncation error that is second order in both time and space.

As a starting point the primary partial differential equation is expressed as

$$\begin{aligned} \dot{C}_T = & D C_{ZZ} + (D_Z - W - WAP - \frac{D}{C^*} C^*_Z) C_Z \\ & + (\frac{D}{C^{*2}} C^{*2}_Z - \frac{1}{C^*} D_Z C^*_Z - \frac{D}{C^*} C^*_{ZZ} - W_Z + \frac{WA}{C^*} C^*_Z) C + G \end{aligned}$$

$$\text{or} \quad C_T = A1 C_{ZZ} + B1 C_Z + C1 C + G. \quad (A1)$$

Here the subscripts pertain to differentiation with respect to the given variable, and species subscripts have been dropped for simplicity.

Addressing next the computation grid in Figure 1, one notes that the grid-space variability can be characterized by an "expansion factor" EF, such that the current grid spacing is equal to the previous grid spacing times EF; that is,

$$DZ(I) = DZ(I-1) * EF(I). \quad (A2)$$

Performing central differencing of some arbitrary variable V about a point I in a manner such that truncation of the corresponding Taylor's series occurs only for terms of order three and above gives

$$\delta_Z(V) = \frac{V(I+1) - [EF(I)]^2 V(I-1) - \{1 - [EF(I)]^2\} V(I)}{EF(I) [1+EF(I)] DZ(I-1)} \quad (A3)$$

and

$$\delta_{ZZ}(V) = 2 \left\{ \frac{V(I+1) + EF(I) V(I-1) - [1+EF(I)] V(I)}{EF(I) [1+EF(I)] [DZ(I-1)]^2} \right\} \quad (A4)$$

Equations (A3) and (A4) can be considered as the basic FDA's for the first and second derivatives, respectively. Owing to the absence of out-lying grid points, these expressions cannot be used at points $I=1$ and $I=N$. Depending upon the nature of the boundary conditions, expansions about these points may or may not be required; in the event that they are, indeed necessary, forward or backward differencing techniques may be applied. Addressing first the bottom boundary, one may perform a forward expansion over the first two grid increments to obtain the form

$$f_Z(V) = \frac{V(1) \{1 - [1+EF(2)]^2\} + V(2) [1+EF(2)]^2 - V(3)}{DZ(1) EF(2) [1+EF(2)]} \quad (A5)$$

Similarly, an FDA for the second derivative can be obtained by forward expansion over the first grid increment:

$$f_{ZZ}(V) = \frac{2}{[DZ(1)]^2} [V(2) - V(1) - DZ(1) V_Z] \quad (A6)$$

Focusing next on the upper boundary one may expand backwards in a similar manner to obtain

$$b_Z(V) = \frac{\xi V(N-1) - V(N-2) + (1-\xi) V(N)}{DZ(N-1) [1 + 1/EF(N-1) - \xi]} \quad (A7)$$

and

$$b_{ZZ}(V) = \frac{2}{[DZ(N-1)]^2} [V(N-1) - V(N) + DZ(N-1) V_Z] \quad (A8)$$

where

$$\zeta = \frac{[1 + EF(N-1)]^2}{[EF(N-1)]^2} \quad (A9)$$

Readdressing the finite-difference grid shown in Figure 1, one can obtain approximations for all values at the virtual, intermediate grid points by taking simple averages of the values at the current and preceeding time steps. For the virtual grid points we have the following relationships:

$$A1(I) = [D(I) + DOLD(I)]/2$$

$$B1(I) = [B1NEW(I) + B1OLD(I)]/2$$

$$C1(I) = [C1NEW(I) + C1OLD(I)]/2$$

With these substitutions the finite-difference approximations can be expressed in the form

$$\delta_T C \doteq A1 \delta_{ZZ} C + B1 \delta_Z C - C1 C + G. \quad (A10)$$

Applying averaged central differences about the intermediate grid points results in the expression

$$AA(I) C(I-1) + BB(I) C(I) + CC(I) C(I+1) = D(I); \quad (A11)$$

where

$$AA(I) = \epsilon \left(\frac{A1}{\gamma} - \frac{\epsilon B1}{2\beta} \right),$$

$$BB(I) = -\frac{1}{DT} - \frac{(1+\epsilon)A1}{\gamma} - \frac{(1-\epsilon^2)B1}{2\beta} + \frac{C1}{2},$$

$$CC(I) = \frac{A1}{\gamma} + \frac{B1}{2\beta},$$

and

$$\begin{aligned} DD(I) = & - AA(I) COLD(I-1) - (BB(I) + A/DT)COLD(I) \\ & - CC(I) COLD(I+1) - G, \end{aligned}$$

where COLD represents the value of C at the previous time step.

Here

$$\begin{aligned} \epsilon &= EF(I) , \\ \beta &= \epsilon(1+\epsilon) DZ(I-1) , \\ \gamma &= \beta DZ(I-1) . \end{aligned}$$

Addressing next the boundary conditions, one notes that different treatments are required depending on whether the conditions are concentration-specified or flux-specified in nature. For concentration-specified conditions one simply supplies the appropriate concentrations at the boundary; this situation removes the need for any solution at the boundary ($I=1$ and or $I=N$), and alters slightly the form of the finite difference expression(s) for the grid point(s) immediately adjacent to the boundary. Thus the appropriate concentration-specified forms are

$$BB(2) C(2) + CC(2) C(3) = DD(2) - AA(2) C(1) \quad (A12)$$

and

$$AA(N-1) C(N-2) + BB(N-1) C(N-1) = DD(N-1) - CC(N-1) C(N) \quad (A13)$$

at the lower and upper boundaries, respectively.

For flux-specified conditions at the grid top, one can write the boundary condition in the form

$$FT = C(N)[W(N) + WAP(N)] - D(N) \left[C_Z - \frac{C(N)}{C^*} \frac{\partial C^*}{\partial Z} \right] \quad (A14)$$

Rearranging this and averaging over the old and current grid points gives

$$\bar{C}_Z = T1 C(N) + T2 \quad (A15)$$

where

$$T1 = \left[\frac{W(N) + WAP(N)}{D(N)} + \frac{C^* Z}{C^*} \right] * .5$$

$$\text{and} \quad T2 = T1OLD \text{ COLD} (N) - \left[\frac{FTOLD}{DOLD (N)} + \frac{FT}{D(N)} \right] * .5$$

T1OLD being the previous time step's value of T1.

Using (A8) to provide a corresponding averaged difference expression for the second derivative, and insertion in (A10) yields the form

$$AA(N) C(N-1) + BB(N) C(N) = DD(N) \quad (A16)$$

where

$$AA(N) = \frac{A1}{[DZ(N-1)]^2}$$

$$BB(N) = - \frac{1}{DT} - AA(N) + \frac{2A1T1}{DZ(N-1)} + B1T1 - \frac{C1}{2}$$

$$\text{and} \quad DD(N) = - \text{COLD}(N) AA(N)$$

$$- \text{COLD}(N) \left[\frac{1}{DT} - AA(N) - \frac{C1}{2} \right]$$

$$- \frac{2A1T2}{DZ(N-1)} - B1T2 - G$$

Corresponding expressions for flux-specified conditions at the grid bottom may be derived in a similar manner. The grid-bottom counterpart of (A14) is

$$FB = C(1) W(1) - D(1) \left[C_Z - \frac{C(1)}{C^*} \frac{\partial C^*}{\partial Z} \right] \quad (A17)$$

which yields the finite-difference expression

$$\bar{C}_Z = S1 C(1) + S2 \quad (A18)$$

where

$$S1 = \left[\frac{W(1)}{D(1)} + \frac{C^* Z}{C^*} \right] * .5$$

and

$$S2 = S1OLD COLD(1) - \left[\frac{FBOLD}{DOLD(1)} + \frac{FB}{D(1)} \right] * .5$$

S1 OLD being the previous time step's value of S1.

Using the forward differencing expression (A6) in a manner similar to that shown above gives

$$BB(1) C(1) + CC(1) C(2) = DD(1) \quad (A19)$$

where

$$CC(1) = \frac{A1}{[DZ(1)]^2}$$

$$BB(1) = -\frac{1}{DT} - CC(1) - \frac{2A1S1}{DZ(1)} + B1S1 - \frac{C1}{2}$$

$$DD(1) = -COLD(1) \left\{ \frac{1}{DT} - CC(1) - \frac{C1}{2} \right\} - COLD(2) CC(1) \\ + 2A1S2/DZ(1) - B1S2 - G$$

APPENDIX B

PLUVIUS PROGRAM LISTING, CLASS II SUBROUTINE LISTINGS AND FLOW CHART

TABLE B.1
PLUVIUS PROGRAM LISTING

```

PROGRAM PLUVIUS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
  COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
  COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
  COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),D
1CDZ(50,3),DDDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),A1(50,4),
2B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
  COMMON /THERMO/ PRESS(50),CS(50)
  DIMENSION AA(100), BB(100), CC(100), DD(100), V(100), S1OLD(25), T
11OLD(25)
  EXTERNAL SAT,DSAT,SSAT

C
C *****
C
C   PROGRAM INITIALIZATION AND INPUT CONTROL
C
C   DEPENDENT VARIABLE CHARACTERISTICS--
C     NCOMP = NO. OF COMPONENTS COMPUTED BY FINITE DIFFERENCING
C     NCNSTR = NO. OF TIME-DEPENDENT VARIABLES COMPUTED
C               BY OTHER MEANS
C
C *****
C
C   READ (5,10) NCOMP,NCNSTR
10  FORMAT (3I5)
   NTOT=NCOMP+NCNSTR
C
C   READ POLLUTANT NAMES
C
C   DO 20 K=1,NTOT
20  READ (5,30) (NAME(LL,K),LL=1,5)
30  FORMAT (5A2)
C
   WRITE (6,50)
   DO 40 K=1,NCOMP
40  WRITE (6,60) K, (NAME(LL,K),LL=1,5)
50  FORMAT (1H1,/,10X,*ONE-DIMENSIONAL MODEL*,1H ,/,10X,*INPUT INFOR
1MATION*,1H ,/,5X,*NAMES OF COMPUTED SPECIES*,/)
60  FORMAT (1H ,5X,I3,5X,5A2)
   IF (NCNSTR.EQ.0) GO TO 90
   WRITE (6,70)

```



```

70  FORMAT (1H ,//,5X,*NAMES OF CONSTRAINED SPECIES*,/)
    N1=NCOMP+1
    N2=NCOMP+'CNSTR
    DO 80 K=N1,N2
80  WRITE (6,60) K, (NAME(LL,K),LL=1,5)
C
C      GRID CHARACTERISTICS
C
C      N = NUMBER OF GRID POINTS IN Z-DIRECTION
C
C      DZ, DZ1, AND DZN = WORKING, LOWEST, AND HIGHEST GRID SPACES
C      EF = GRID EXPANSION FACTOR (DZ(I) = DZ(I-1)*EF(I))
C      T = MODEL TIME LIMIT FOR COMPUTATION
C      DT = INCREMENTAL TIME SPACING
C      LTD = TOTAL NUMBER OF "DIFFUSION REGIEMES"
C
C
90  READ (5,100) N,NPRINT,LTD,DZ,T,DT
100  FORMAT (3I5,3E10.3)
    WRITE (6,110)
110  FORMAT (1H ,//,*****1H ,//,10X,*
1CONSTANT UNITS FOR INPUT AND OUTPUT ARE*,* GIVEN AS FOLLOWS:*,1H
2 ,//,10X,*LENGTH: CENTIMETERS*,1H ,//,10X,*MASS: GRAM-MOLES*,1H
3,//,10X,*TIME: SECONDS*,1H ,//,10X,*FORCE: NEWTONS*,1H ,//,10X,*
4TEMPERATURE: DEGREES KELVIN*,1H ,//,*****
5******)
    DTOLD=DT
    CALL GRDSET
    WRITE (6,120)
    WRITE (6,130) DT,NPRINT
    WRITE (6,140)
120  FORMAT (1H1,20X,*GRID CHARACTERISTICS*,//)
130  FORMAT (1H ,10X,*TIME STEP = *,E10.4,* SECONDS*,10X,* PRINTING INT
1ERVAL = *,I5,* TIME STEPS*)
140  FORMAT (1H ,//,20X,*VERTICAL GRID FEATURES*,//,10X,* INDEX      H
1EIGHT      *,*EXPANSION FACTOR*,//)
    DO 150 II=1,N
    III=N-II+1
150  WRITE (6,160) III,Z(III),EF(III)
160  FORMAT (1H ,10X,I5,5X,E10.4,6X,F10.4)
C
C      FIXED VS. VARIABLE MODE OPERATIONS (SUBROUTINES
C      INTERROGATED INITIALLY FOR BOTH MODES, BUT SUBSEQUENTLY
C      AT EACH TIME STEP ONLY FOR VARIABLE MODE)
C
    READ (5,170) IGEN,IDIFF,ITEMP,IPRESS,ISETTL,IBOUND,IVERT,IFOG
170  FORMAT (8I5)
C

```

```

C      THE TERMS IN THE ABOVE READ STATEMENT PERTAIN TO
C      1) THE VOLUMETRIC GENERATION SUBROUTINE
C      2) THE DIFFUSIVITY SUBROUTINE
C      3) THE TEMPERATURE SUBROUTINE
C      4) THE DENSITY SUBROUTINE
C      5) THE RELATIVE VERTICAL VELOCITY SUBROUTINE
C      6) THE BOUNDARY CONDITION SUBROUTINE
C      7) THE TERRAIN SUBROUTINE
C      8) CONSTRAINED NON-CLOUD CONDITIONS
C      VALUE 1 PERTAINS TO VARIABLE MODE; VALUE 2
C      PERTAINS TO FIXED MODE.
C
C      BOUNDARY CONDITION CHARACTERISTICS
C      ITOP(K) SPECIFIES TYPE OF BOUNDARY CONDITION FOR
C      COMPONENT K ON TOP OF GRID
C      IBOT(K) SPECIFIES TYPE OF BOUNDARY CONDITION FOR
C      COMPONENT K ON BOTTOM OF GRID
C      VALUE 1 PERTAINS TO FLUX-SPECIFIED CONDITION; VALUE 2
C      PERTAINS TO CONCENTRATION-SPECIFIED CONDITION.
C
      DO 180 K=1,NCOMP
180    READ (5,190) ITOP(K),IBOT(K),LDIF(K)
190    FORMAT (3I5)
      WRITE (6,200)
200    FORMAT (1H1,/,10X,*BOUNDARY CONDITION STATUS*,1H,10X,*(1 DENOTES
1FLUX-SPECIFIED AND 0 DENOTES *,*CONCENTRATION SPECIFIED)*,1H ,//10
2X,*COMPONENT*,11X,*TOP*,10X,*BOTTOM*,/)
      DO 210 K=1,NCOMP
210    WRITE (6,220) (NAME(LL,K),LL=1,5),ITOP(K),IBOT(K)
220    FORMAT (1H ,10X,5A2,10X,I1,13X,I1)
C
C      *****
C
C      BEGIN GRID INITIALIZATION OPERATIONS
C
      NM1=N-1
      INDEX=1+ITEMP
      ITERM=0
      IPRINT=0
      TCLOCK=0
      ISLDER=IDIFF+ITEMP+IVERT
      ISLSET=ISLDER+ISETTL
      GLEVL=0.
      INIT=1
      CALL DIFF (INIT)
      CALL VERT (TCLOCK)
      CALL PRESSF
      CALL TEMPF (INIT,IFOG)
      CALL CONC (INIT)
      CALL GEN (INIT)

```

```

CALL SETTLE (INIT)
CALL BOUND (INIT)
CALL PRINT
CALL DERIV (INIT)
DO 230 K=2,NCOMP
L=LDIF(K)
LM1=L-1
CALL SETVEC (K,INIT)
S1OLD(K)=(W(1,K)/D(1,L)+DCDZ(1,LM1))*0.5
230 T1OLD(K)=((W(N,K)+WAP(N))/D(N,L)+DCDZ(N,LM1))*0.5
IF (INDEX.EQ.1) GO TO 240
TDOLD=7.176*CS(N)*D(N,1)
T1OLD(1)=WAP(N)/TDOLD
SDOLD=FB(1)/(7.176*CS(1)*D(1,1))
240 INIT=0
C
C      START MAJOR COMPUTATION LOOP. INITIALIZE GRIDS
C      FOR NEXT COMPUTATION STEP.
C
250 DO 260 I=1,N
DO 260 L=1,LTD
260 DOLD(I,L)=D(I,L)
RDT=1./DT
DO 280 K=1,NCOMP
FBOLD(K)=FB(K)
280 FTOLD(K)=FT(K)
IF (IGEN.EQ.0) GO TO 290
CALL GEN (INIT)
290 IF (ISETTL.EQ.0) GO TO 300
CALL SETTLE (INIT)
300 IF (IPRESS.EQ.0) GO TO 310
CALL PRESSF
310 IF (IDIFF.EQ.0) GO TO 320
CALL DIFF (INIT)
320 IF (IBOUND.EQ.0) GO TO 330
CALL BOUND (INIT)
330 TCLOCK=TCLOCK+DT
IF (IVERT.EQ.0) GO TO 340
CALL VERT (TCLOCK)
C
C      NOW NEW AND OLD MATRICES ARE READY FOR THE NEXT STEP
C      OF FINITE DIFFERENCE COMPUTATION. SOLUTIONS WILL BE
C      PERFORMED SEQUENTIALLY FOR EACH COMPONENT K. START
C      BY CREATING AA, BB, CC, AND DD VECTORS FOR INTERNAL
C      GRID FOR KTH COMPONENT
C
C      CREATE SPECIES DEPENDENT COEFFICIENTS B1,C1
C

```

```

340   DO 510 K=1,NCOMP
      L=LDIF(K)
      LM1=L-1
      IF (ITEMP.EQ.0.AND.K.EQ.1) GO TO 510
      IF (K.EQ.1TEMP) GO TO 370
      IF (K.NE.2) GO TO 350
      IF (ISLDER.EQ.0) GO TO 350
      CALL DERIV (INIT)
350   DO 360 I=1,N
      B1(I)=BNEW(I,K)
360   C1(I)=CNEW(I,K)
      IF (ISLSET.EQ.0) GO TO 380
      CALL SETVEC (K,INIT)
      GO TO 380
370   CALL TEMPF (INIT,IFOG)
380   DO 390 I=2,NM1
      W1=A1(I,L)/GAMMA(I)
      W2=B1(I)*.5/BETA(I)
      AA(I)=EF(I)*(W1-EF(I)*W2)
      BB(I)=-RDT-EFP1(I)*W1+EFSQM1(I)*W2+.5*C1(I)
      CC(I)=W1+W2
390   DD(I)=-AA(I)*C(I-1,K)-(BB(I)+2.*RDT)*C(I,K)-CC(I)*C(I+1,K
      1)-G(I,K)

C
C      MAKE APPROPRIATE ENTRIES IN VECTORS FOR BOUNDARY CONDITIONS;
C      START WITH TOP CONDITION
C
      IF (ITOP(K).EQ.0) GO TO 430
C
C      FLUX-SPECIFIED CONDITIONS
C
      IF (ITEMP.EQ.K) GO TO 400
      T1=((W(N,K)+WAP(N))/D(N,L)+DCDZ(N,LM1))* .5
      T2=T1OLD(K)*C(N,K)-(FTOLD(K)/DOLD(N,L)+FT(K)/D(N,L))* .5
      GO TO 420
400   DFACT=0.
      IF (IFOG.EQ.0) GO TO 410
C
C#####
C
      CFOG=C(N,2)-SAT(C(N,1))
C
C#####
      TD=7.176*CS(N)*D(N,1)
      T1=.5*WAP(N)/(1.3829*D(N,1))
      IF (CFOG.LT.0.) GO TO 410
      DFACT=10800.*DSAT(C(N,1))/CS(N)
410   DTAUDZ=-.000701/(7.176+DFACT)
      T2=.5*(T1OLD(1)*C(N,1)-FTOLD(1)/TDOLD-FT(1)/TD)+DTAUDZ
      TDOLD=TD

```

```

420  AA(N)=A1(N,L)/(DZN*DZN)
      BB(N)=-RDT-AA(N)+2.*A1(N,L)*T1/DZN+B1(N)*T1+.5*C1(N)
      DD(N)=-C(N-1,K)*AA(N)-C(N,K)*(RDT-AA(N)+.5*C1(N))-2.*A1(N,L)
      1*T2/DZN-B1(N)*T2-G(N,K)
      T1OLD(K)=T1
      GO TO 440

C
C      CONCENTRATION-SPECIFIED CONDITIONS
C
430  DD(N-1)=DD(N-1)-C(N,K)*CC(N-1)
C
C      PROCEED TO BOTTOM CONDITION
C
440  IF (IBOT(K).EQ.0) GO TO 480
C
C      FLUX-SPECIFIED CONDITIONS
C
      IF (ITEMP.EQ.K) GO TO 450
      S1=(W(1,K)/D(1,L)+DCDZ(1,LM1))*.5
      S2=S1OLD(K)*C(1,K)-(FBOLD(K)/DOLD(1,L)+FB(K)/D(1,L))*.5
      GO TO 470
450  DFACT=0.
      IF (IFOG.EQ.0) GO TO 460
C
C#####
C
      CFOG=C(I,2)-SAT(C(1,1))
C
C#####
C
      SD=FB(1)/(7.176*CS(1)*D(1,1))
      IF (CFOG.LT.0.) GO TO 460
      DFACT=10800.*DSAT(C(1,1))/CS(1)
460  DTAUDZ=-.000701/(7.176+DFACT)
      S2=-.5*(SD+SDOLD)+DTAUDZ
      S1=0.
      SDOLD=SD
470  CC(1)=A1(1,L)/(DZ1*DZ1)
      BB(1)=-RDT-CC(1)-2.*A1(1,L)*S1/DZ1+B1(1)*S1+.5*C1(1)
      DD(1)=-C(1,K)*(RDT-CC(1)+C1(1)/2.)-C(2,K)*CC(1)+2.*A1(1,1)*S
      12/DZ1-B1(1)*S2-G(1,K)
      S1OLD(K)=S1
      GO TO 490

C
C      CONCENTRATION SPECIFIED CONDITIONS
C
480  DD(2)=DD(2)-C(1,K)*AA(2)
C
C      SOLVE MATRIX FOR COMPONENT K
C

```

```

490  IUP=N-1+ITOP(K)
      IDN=2-IBOT(K)
      DO 270 KK=1,NTOT
      DO 270 I=1,N
270  COLD(I,KK)=C(I,KK)
      V(1)=C(1,K)
      V(N)=C(N,K)
      CALL TRIDAG (IDN,IUP,AA,BB,CC,DD,V)
      DO 500 I=1,N
500  C(I,K)=V(I)
510  CONTINUE
      IPRINT=IPRINT+1
      CALL CONC (INIT)
      DTOLD=DT
      CALL TIMESP
      CALL COLLECT (ITERM)
      IF (IPRINT.LT.NPRINT) GO TO 520
      CALL PRINT
      IPRINT=0
520  IF (TCLOCK.LT.T) GO TO 250
      ITERM=1
      CALL COLLECT (ITERM)
      STOP
      END

```

TABLE B.2

LISTING OF CLASS II SUBROUTINE PRINT

```

SUBROUTINE PRINT
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),D
1CDZ(50,3),DDDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),A1(50,4),
2B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)
NTOT=NCOMP+NCNSTR
DO 70 IDUM=1,2
NSET=0
NDUM=NCOMP
NGAP=NCOMP
IF (IDUM.EQ.1) GO TO 10
IF (NCNSTR.EQ.0) GO TO 70
NDUM=NTOT
NSET=NCOMP
NGAP=NTOT-NSET
10 NPAGE=(NGAP-1)/10+1
DO 60 II=1,NPAGE
IF (IDUM.EQ.2) GO TO 1
WRITE (6,30) TCLOCK,II,DT
GO TO 2
1 WRITE(6,35) TCLOCK,II,DT
2 WRITE (6,20) GLEVL,DZDT
20 FORMAT (1H ,//,* LOWER BOUNDARY POSITION ABOVE DATUM = *,E12.4,*
1 CURRENT ASCENT RATE = *,E12.4)
30 FORMAT (1H1,//,20X,* COMPUTED RESULTS FOR *,E10.4,* SECONDS OF MO
1DEL TIME, PAGE *,I5,//,* CURRENT TIME INCREMENT IS*,E10.4,*SECO
2S*)
35 FORMAT (1H1,//,20X,* CONSTRAINED RESULTS FOR *,E10.4,* SECONDS OF
1 MODEL TIME, PAGE *,I5,//,* CURRENT TIME INCREMENT IS*,E10.4,*SECO
2NDS*)
ILO=10*II-9+NSET
INT=10
ITEST=NDUM-ILO+1
IF (ITEST.LT.10) INT=ITEST
IHI=ILO+INT-1
WRITE (6,40) ((NAME(LL,K),LL=1,5),K=ILO,IHI)
40 FORMAT (1H ,//,10X,* HEIGHT *,10(1X,5A2),/)
DO 60 J=1,N
I=N+1-J
ZZ=Z(I)+GLEVL
WRITE (6,50) ZZ,(C(I,K),K=ILO,IHI)

```

```

50  FORMAT (1H ,10X,E10.4,10(1X,E10.3))
60  CONTINUE
70  CONTINUE
    WRITE (6,80)
80  FORMAT (1H1,/,20X,*PRESSURE PROFILE*,/,1H ,10X,*  HEIGHT  PRES
1SURE  AIR CONC*,/)
    DO 90 J=1,N
      I=N+1-J
      ZZ=Z(I)+GLEVL
90  WRITE (6,100) ZZ,PRESS(I),CS(I)
100 FORMAT (1H ,10X,E10.4,1X,E10.4,1X,E10.4)
    RETURN
    END

```


TABLE B.3

LISTING OF CLASS II SUBROUTINE TRIDAG

```

SUBROUTINE TRIDAG (IF,L,A,B,C,D,V)
C
C      SUBROUTINE FOR SOLVING A SYSTEM OF LINEAR SIMULTANEOUS
C      EQUATIONS HAVING A TRIDIAGONAL COEFFICIENT MATRIX.
C      THE EQUATIONS ARE NUMBERED FROM IF THROUGH L, AND THEIR
C      SUB-DIAGONAL, DIAGONAL, AND SUPER-DIAGONAL COEFFICIENTS
C      ARE STORED IN THE ARRAYS A, B, AND C.  THE COMPUTED
C      SOLUTION VECTOR V(IF)...V(L) IS STORED IN THE ARRAY V.
C
      DIMENSION A(100), B(100), C(100), D(100), V(100), BETA(101), GAMMA
1(101)
C
C      COMPUTE INTERMEDIATE ARRAYS BETA AND GAMMA
C
      BETA(IF)=B(IF)
      GAMMA(IF)=D(IF)/BETA(IF)
      IFP1=IF+1
      DO 10 I=IFP1,L
      BETA(I)=B(I)-A(I)*C(I-1)/BETA(I-1)
10  GAMMA(I)=(D(I)-A(I)*GAMMA(I-1))/BETA(I)
C
C      COMPUTE FINAL SOLUTION VECTOR V
C
      V(L)=GAMMA(L)
      LAST=L-IF
      DO 20 K=1,LAST
      I=L-K
20  V(I)=GAMMA(I)-C(I)*V(I+1)/BETA(I)
      RETURN
      END

```

TABLE B.4

LISTING OF CLASS II SUBROUTINE DERIV

```

SUBROUTINE DERIV (INIT)
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),D
1CDZ(50,3),DDDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),A1(50,4),
2B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)

SUBROUTINE COMPUTES THOSE VARIABLES WHICH ARE INDEPENDENT
OF SPECIES AND ARE REQUIRED FOR SUBSEQUENT USE BY THE MAIN
PROGRAM AND SUBROUTINE SETVEC. CALLED ONCE PER TIME STEP,
UNLESS BYPASSED.

NM1=N-1
DO 50 L=2,LTD
LM1=L-1
DO 10 I=2,NM1
DDDZ(I,LM1)=(D(I+1,L)-EFSQ(I)*D(I-1,L)+EFSQM1(I)*D(I,L))/BETA(I)
DCDZ(I,LM1)=(CS(I+1)-EFSQ(I)*CS(I-1)+EFSQM1(I)*CS(I))/BETA(I)
DSCDZ(I,LM1)=2.*(CS(I+1)+EF(I)*CS(I-1)-EFP1(I)*CS(I))/GAMMA(I)
DCDZ(I,LM1)=DCDZ(I,LM1)/CS(I)
DSCDZ(I,LM1)=D(I,L)*DCDZ(I,LM1)*DCDZ(I,LM1)-(DDDZ(I,LM1)-DZDT-WAP(
1I))*DCDZ(I,LM1)-D(I,L)*DSCDZ(I,LM1)/CS(I)
DDDZ(I,LM1)=DDDZ(I,LM1)-D(I,L)*DCDZ(I,LM1)-WAP(I)
DDDZ(N,LM1)=(ZETA*D(N-1,L)-D(N-2,L)+(1.-ZETA)*D(N,L))/EPS
DCDZ(N,LM1)=(ZETA*CS(N-1)-CS(N-2)+(1.-ZETA)*CS(N))/EPS
DSCDZ(N,LM1)=2.*(CS(N-1)-CS(N)+DZN*DCDZ(N,LM1))/(DZN*DZN)
DCDZ(N,LM1)=DCDZ(N,LM1)/CS(N)
DSCDZ(N,LM1)=D(N,L)*DCDZ(N,LM1)**2-(DDDZ(N,LM1)-DZDT-WAP(N))*DCDZ(
1N,LM1)-D(N,L)*DSCDZ(N,LM1)/CS(N)
DDDZ(N,LM1)=DDDZ(N,LM1)-D(N,L)*DCDZ(N,LM1)-WAP(N)
DDDZ(1,LM1)=(D(1,L)*(1.-EP1SQ)+D(2,L)*EP1SQ-D(3,L))/SIGMA
DCDZ(1,LM1)=(CS(1)*(1.-EP1SQ)+CS(2)*EP1SQ-CS(3))/SIGMA
DSCDZ(1,LM1)=2.*(CS(2)-CS(1)-DZ1*DCDZ(1,LM1))/(DZ1*DZ1)
DCDZ(1,LM1)=DCDZ(1,LM1)/CS(1)
DSCDZ(1,LM1)=D(1,L)*DCDZ(1,LM1)**2-(DDDZ(1,LM1)-DZDT)*DCDZ(1,LM1)-
1D(1,L)*DSCDZ(1,LM1)/CS(1)
DDDZ(1,LM1)=DDDZ(1,LM1)-D(1,L)*DCDZ(1,LM1)
IF (INIT.NE.1) GO TO 30
DO 20 I=1,N
A1(I,L)=D(I,L)
GO TO 50
30 DO 40 I=1,N
40 A1(I,L)=(D(I,L)+DOLD(I,L))/2.
50 CONTINUE
RETURN
END

```

TABLE B.5

LISTING OF CLASS II SUBROUTINE SETVEC

```

SUBROUTINE SETVEC (K,INIT)
COMMON /GRID/ N,DZ,DT,DZ1,DZN,EF(50),EFSQ(50),EFP1(50),EFSQM1(50),
1BETA(50),GAMMA(50),Z(50),ITOP(25),IBOT(25),EPS,ZETA,SIGMA,EP1SQ,DZ
2DT,WAP(50),GLEVL,LDIF(25),LTD,DTOLD
COMMON /NAMES/ NAME(5,25),NCOMP,NCNSTR
COMMON C(50,25),COLD(50,25),G(50,25),D(50,4),DOLD(50,4),W(50,25),D
1CDZ(50,3),DDDZ(50,3),DSCDZ(50,3),BNEW(50,25),CNEW(50,25),A1(50,4),
2B1(50),C1(50),FTOLD(25),FT(25),FBOLD(25),FB(25),TCLOCK
COMMON /THERMO/ PRESS(50),CS(50)
NM1=N-1

C
C      SUBROUTINE CREATES VECTORS B1 AND C1 FOR SUBSEQUENT USE BY
C      MAIN PROGRAM.  CALLED ONCE FOR EACH SPECIES K, UNLESS BYPASSED.
C

  IF (INIT.EQ.1) GO TO 20
  DO 10 I=1,N
    B1(I)=BNEW(I,K)
10   C1(I)=CNEW(I,K)
20   L=LDIF(K)
    LM1=L-1
    DO 30 I=2,NM1
      DWDZ=(W(I+1,K)-EFSQ(I)*W(I-1,K)+EFSQM1(I)*W(I,K))/BETA(I)
      BNEW(I,K)=DDDZ(I,LM1)-W(I,K)
      CNEW(I,K)=DSCDZ(I,LM1)-DWDZ
      IF (INIT.EQ.1) GO TO 30
      B1(I)=(B1(I)+BNEW(I,K))/2.
      C1(I)=(C1(I)+CNEW(I,K))/2.
30   CONTINUE
      IF (ITOP(K).NE.1) GO TO 40
      DWDZ=(ZETA*W(N-1,K)-W(N-2,K)+(1.+ZETA)*W(N,K))/EPS
      BNEW(N,K)=DDDZ(N,LM1)-W(N,K)
      CNEW(N,K)=DSCDZ(N,LM1)-DWDZ
      IF (INIT.EQ.1) GO TO 40
      B1(N)=(B1(N)+BNEW(N,K))/2.
      C1(N)=(C1(N)+CNEW(N,K))/2.
40   IF (IBOT(K).NE.1) GO TO 50
      DWDZ=(W(1,K)*(1.-EP1SQ)+W(2,K)*EP1SQ-W(3,K))/SIGMA
      BNEW(1,K)=DDDZ(1,LM1)-W(1,K)
      CNEW(1,K)=DSCDZ(1,LM1)-DWDZ
      IF (INIT.EQ.1) GO TO 50
      B1(1)=(B1(1)+BNEW(1,K))/2.
      C1(1)=(C1(1)+CNEW(1,K))/2.
50   IF (INIT.NE.1) GO TO 70
      DO 60 I=1,N
        B1(I)=BNEW(I,K)
60   C1(I)=CNEW(I,K)
70   RETURN
      END

```

FIGURE B.1

FLOWCHART FOR MAIN PROGRAM PLUVIUS

PROGRAM INITIALIZATION AND INPUT CONTROL

MAJOR COMPUTATION LOOP

B-13

DISTRIBUTION

<u>No. of Copies</u>		<u>No. of Copies</u>	
	A. A. Churm DOE Patent Division 9800 S. Cass Avenue Argonne, IL 60439		A. Lazrus National Center for Atmospheric Research P.O. Box 3000 Boulder, CO 80302
	D. S. Ballantine Mail Stop E-201, Germantown U.S. Department of Energy Washington, D.C. 20545		J. Wyngaard National Center for Atmospheric Research P.O. Box 3000 Boulder, CO 80302
10	R. W. Beadle U.S. Department of Energy Mail Stop E-201 Washington, D.C. 20545		J. Miller NOAA 8060 13th Street Silver Spring, MD 20910
	D. H. Slade Div. of Biomedical & Env. Res. U.S. Department of Energy Washington, D.C. 20545		E. Cowling North Carolina State University Box 5397 Raleigh, NC 27607
27	DOE Technical Information Center		
10	J. D. Shannon Argonne National Laboratory 9700 South Cass Bldg. 181 Argonne, IL 60439		R. Linthurst North Carolina State University Box 6186 Raleigh, NC 27650
	J. H. Gibson Colorado State University Fort Collins, CO 80523		J. Bachman U.S. Environmental Protection Agency MD-12 Research Triangle Park, NC 27711
	G. Hilst Electric Power Research Institute P.O. Box 10412 Palo Alto, CA 94304		J. Durham U.S. Environmental Protection Agency MS-57 Research Triangle Park, NC 27711
	R. G. Semonin Illinois State Water Survey P.O. Box 232 Urbana, IL 61801		G. Foley U.S. Environmental Protection Agency 401 M Street S.W. RD-682 Washington, D.C. 20460
	M. MacCracken Lawrence Livermore Laboratory P.O. Box 808 Livermore, CA 94550	10	W. Keith U.S. Environmental Protection Agency 401 M Street S.W. RD-682 Washington, D.C. 20460

No. of
Copies

No. of
Copies

10 N. S. Laulainen
U.S. Environmental Protection Agency
401 M. Street S.W. RD-682
Washington D.C. 20460

R. Wilhour
U.S. Environmental Protection Agency
2000 S.W. 35th Street
Corvallis, OR 97330

J. Galloway
University of Virginia
Clark Hall
Charlottesville, VA 22903

R. Husar
Washington University
St. Louis, MO 63130

86 Pacific Northwest Laboratory

A. Alkezweeny
C. Berkowitz
C. Doran
R. Easter
J. Hales (73)
B. Scott
L. Wendell
Publishing Coordination (2)
Technical Information (5)

FOREIGN

L. Barrie
Atmospheric Environment Service
4905 Dufferin Street
Downsview, Ontario
Canada M3H 5T4

S. Slania
Netherlands Energy Research
Foundation Research Centre
3. Westerduinweg
Petten (NH)
Netherlands

L. Granat
Department of Meteorology
University of Stockholm
S-106 91 Stockholm
Sweden

ONSITE

DOE Richland Operations Office

H. E. Ransom