

ISORROPIA v1.7
REFERENCE MANUAL

Copyright 1996-2006
University of Miami, Carnegie Mellon
University, Georgia Institute of Technology

TABLE OF CONTENTS

PREFACE

ACKNOWLEDGMENTS

1. INTRODUCTION.....	1
2. ISORROPIA OVERVIEW	2
2.1 Introduction.....	2
2.2 Filing system.....	3
2.3 The interactive driver	5
2.3.1 Introduction.....	5
2.3.2 Configuration file ISRPIA.CNF	5
2.3.3 Batch mode input files	7
2.3.4 Spreadsheet output format.....	8
2.3.5 Creating customized output.....	10
3. ISORROPIA PROGRAMMING REFERENCE	11
3.1 SUBROUTINE ISORROPIA.....	11
3.2 SUBROUTINE SETPARAM	14
3.3 SUBROUTINE GETPARAM	16
3.4 SUBROUTINE ISERRINF.....	17
3.5 SUBROUTINE ERRSTAT	18
3.6 SUBROUTINE ISORINF	19
4. ISORROPIA STRUCTURE	21
5. SUBROUTINE REFERENCE.....	30
6. DESCRIPTION OF GLOBAL VARIABLES	67
7. EXAMPLES OF USING ISORROPIA.....	73

7.1	Interactive demo	73
7.2	Batch mode demo	75
7.3	Generating customized output	76
8.	EXAMPLES OF PROGRAMMING ISORROPIA	77
8.1	Using the full interface	77
8.2	Incorporation into air quality (or global) model	78
9.	REFERENCES.....	80
APPENDIX A:	CHANGES BETWEEN VERSIONS OF ISORROPIA	81
	Changes between versions 1.5 and 1.7	81
	Changes between versions 1.4 and 1.5	82
	Changes between versions 1.3 and 1.4	82
	Changes between versions 1.1 and 1.3	83
	Changes between versions 1.0 and 1.1	85
APPENDIX B:	OLDER INTERFACES OF ISORROPIA.....	86
	Interface for versions 1.1 to 1.6	86
	Interface for version 1.0.....	86
APPENDIX C:	COMPILING ISORROPIA	89
	Compiling with interactive driver	89
	Compiling with air quality (or global) model	90

Preface

For the most recent copies of the ISORROPIA computer code and manual, for reporting errors, for subscribing to the ISORROPIA mail list, or for any other question please contact one of the following persons:

Name: Christos Fountoukis

Address: 311 Ferst Drive NW, School of Chemical & Biomolecular Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0100, USA

Telephone: (404) 385-4695

e-mail: Christos.Fountoukis@chbe.gatech.edu

Name: Athanasios Nenes

Address: 311 Ferst Drive NW, Schools of Earth and Atmospheric Sciences and Chemical & Biomolecular Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0100, USA

Telephone: (404) 894-9225

e-mail: nenes@eas.gatech.edu

Name: Dr. Spyros Pandis

Address: Department of Chemical Engineering and Engineering and Public Policy, Carnegie Mellon University, Pittsburgh, PA, 15213, USA

Telephone: (412) 268-3531

e-mail: spyros@andrew.cmu.edu

Name: Dr. Christodoulos Pilinis

Address: Environmental Science Department, University of the Aegean, Karadoni 17, Mytilene, GR - 81 100, GREECE

Telephone: +30-251-36233

e-mail: xpil@aegean.gr

Acknowledgments

The development of the first release of the code (version 1.0) was done with support from the Environmental Protection Agency under grant R-824793010, from the National Science Foundation under grant ATM-9625718, and from the ONR studentship by grant N000149510807.

Versions 1.1 to 1.6 were developed with the contributions of many people. In particular we would like to acknowledge the contributions (in alphabetical order): Asif Ansari, Veronique Bouchet, Prakash Bhawe, Bill Hutzell, Kevin Capaldo, Bonyoung Koo, Sonia Kreidenweis, Paul Makar, Federico San Martini, Chris Nolte, Betty Pun, Armistead Russell, Uma Shankar, Jason West, Douglas Waldron, Ashraf Zaakey, Yang Zhang. The feedback and bug reports from many other individuals using the code is greatly appreciated.

Version 1.7 was developed with support from the National Oceanic and Atmospheric Administration (NOAA) under contract NMRAC000-5-04017. We would also like to acknowledge the contributions of Prakash Bhawe and Chris Nolte.

1. INTRODUCTION

This manual documents the usage, structure and programming interface of the ISORROPIA inorganic aerosol thermodynamic equilibrium model. This manual can serve either as a user's guide or a programmer's reference guide. All the information provided pertains to version 1.7 of ISORROPIA (dated March 26, 2007). In subsequent sections, the program structure and the file system of ISORROPIA are described in detail. Then, the program interface subroutines and global variables allocated in the COMMON blocks are presented. Finally, a few examples displaying the usage of ISORROPIA are given. One example demonstrates its usage as an independent program, while another displays how to call ISORROPIA from within another program. Finally, another example outlines the basic principles for embedding ISORROPIA into larger air quality (or global) models.

2. ISORROPIA OVERVIEW

2.1 Introduction

ISORROPIA was originally developed at the Division of Marine and Atmospheric Chemistry of the Rosenstiel School of Marine and Atmospheric Science, University of Miami. The objectives were to develop an aerosol thermodynamics model that balances accuracy with computational efficiency. The complete theory of ISORROPIA, together with a detailed description of the equations solved, the activity coefficient calculation methods and the computational algorithms used is presented elsewhere (Nenes *et al.*, 1998a,b). The performance and advantages of ISORROPIA over the usage of other thermodynamic equilibrium codes is assessed by Nenes *et al.* (1998b) and Ansari and Pandis (1999a).

ISORROPIA can solve two classes of problems:

- *Forward problems*, in which known quantities are temperature, relative humidity and the total (i.e. gas+aerosol) concentrations of NH_3 , H_2SO_4 , Na, HCl and HNO_3 . The concentrations of these precursors refer to the total amount in the gas and aerosol phases.
- *Reverse problems*, in which known quantities are temperature, relative humidity and the aerosol phase concentrations of NH_3 , H_2SO_4 , Na, HCl and HNO_3 . These concentrations refer to the total amount of these precursors in the aerosol phase. For example, under certain conditions, one could find NH_4^+ present in both the aqueous and solid phases. The NH_3 would then be the sum of ammonium found in both phases of the aerosol.

For both problem classes, ISORROPIA predicts which species can exist in the gas and aerosol phase and calculates their concentrations at chemical equilibrium. The user can specify

the aerosol to be either in a thermodynamically stable state, where salts precipitate if saturation is exceeded, or in a metastable state, where salts do not precipitate under supersaturated conditions. The aerosol in the first case can be solid, liquid or both, while the second case is always an aqueous solution.

The source code of ISORROPIA is written primarily in FORTRAN 77. The only extensions to the ANSI standard implemented were the usage of `INCLUDE` statements, trailing comments, and variable names with length larger than 6 characters (but not larger than 9). These extensions are supported by the majority of FORTRAN 77 compilers, and are also included in the Fortran 90 standard, so they should not pose any portability problems. The model has extensively been tested on many platforms (DEC Alpha, HP and IBM RISC workstations under UNIX, Intel-based PC's under MS-DOS and Windows 95) and compilers (both FORTRAN 77 and Fortran 90). The results during these tests were found to be independent of the system used.

2.2 Filing system

The filing system of ISORROPIA is presented in Figure 1. Four files, `ISOCOM.FOR`, `ISOFWD.FOR`, `ISOREV.FOR` and `ISRPIA.INC` contain the ISORROPIA source code. The `.FOR` files consist the program segments (subroutines, functions, block data), while the `.INC` file contains the definition of the `COMMON` blocks. Of these files, the three `.FOR` are those that have to be compiled. The `.INC` file is automatically read by the compiler during compilation (via `INCLUDE` statements) and should not be separately compiled.

Apart from the model source code, a driver is provided that allows the interactive execution of ISORROPIA. This driver is contained in files `MAIN.FOR`, `MAIN.INC`, and has been designed to be as versatile as possible. Input can be provided either directly from the

CHAPTER 2: ISORROPIA OVERVIEW

temperature (Kelvin). The concentration units, and the problem type solved are those specified in the header. Output is controlled by the parameters specified in `ISRPIA.CNF`.

List 2: Sample batch mode input file.

```
1. Input units (0=umol/m3, 1=ug/m3)
2. 0
3.
4. Problem type (0=forward, 1=reverse); Phase state (0=solid+liquid, 1=metastable)
5. 0, 0
6.
7. NH4-SO4 system case
8. Na SO4 NH3 NO3 Cl RH TEMP
9. 0 0.379 0.758 0 0 0.90 298.0
```

2.3.4 Spreadsheet output format

The spreadsheet data file is a standard ASCII file, that uses double quotes (i.e., “”) as text delimiters and regular spaces as numerical delimiters. The first line is a header which provides a label for each data column. Whenever ISORROPIA completes a run, one line of data is written with all the information regarding the input and equilibrium concentrations. Table 1 contains a detailed description of each column. All concentrations are expressed in $\mu\text{g m}^{-3}$ air.

CHAPTER 2: ISORROPIA OVERVIEW

Table 1: Spreadsheet file format

Column #	Quantity (concentrations in $\mu\text{g m}^{-3}$ air)
1	Total sodium (gas+aerosol) expressed as equivalent Na
2	Total sulfate (gas+aerosol) expressed as equivalent H_2SO_4
3	Total ammonium (gas+aerosol) expressed as equivalent NH_3
4	Total nitrate (gas+aerosol) expressed as equivalent HNO_3
5	Total chloride (gas+aerosol) expressed as equivalent HCl
6	Ambient relative humidity, expressed on a (0,1) scale
7	Ambient temperature, K
8	Gas phase NH_3
9	Gas phase HCl
10	Gas phase HNO_3
11	Solid phase NaCl
12	Solid phase NaNO_3
13	Solid phase Na_2SO_4
14	Solid phase NaHSO_4
15	Solid phase NH_4Cl
16	Solid phase NH_4NO_3
17	Solid phase $(\text{NH}_4)_2\text{SO}_4$
18	Solid phase NH_4HSO_4
19	Solid phase $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$
20	Aqueous phase H^+
21	Aqueous phase Na^+
22	Aqueous phase NH_4^+
23	Aqueous phase Cl^-
24	Aqueous phase SO_4^{2-}
25	Aqueous phase HSO_4^-
26	Aqueous phase NO_3^-
27	Total aerosol ammonium (solid + aqueous) expressed as equivalent NH_3
28	Total aerosol chloride (solid + aqueous) expressed as equivalent HCl
29	Total aerosol nitrate (solid + aqueous) expressed as equivalent HNO_3
30	Aerosol water content
31	Aerosol aqueous phase mass (water+dissolved salts)
32	Aerosol solid phase mass
33	ISORROPIA subcase

2.3.5 Creating customized output

Customized data output can be generated by using SUBROUTINE USROUT (List 3), which is found in file MAIN.FOR. The driver calls USROUT in three phases of a session. The formal INTEGER argument ICODE informs the programmer which phase USROUT is called from. First (ICODE=1), USROUT is called right before the driver performs any runs, so the modeler can OPEN any files, generate any initial output, etc. Then (ICODE=2) it is called right after a run, so any run-specific output can be generated. Finally, (ICODE=3) USROUT is called once before the driver terminates, and allows for any final output. By default, USROUT does not generate anything, but provides a basic BLOCK IF structure on which the modeler “builds” upon. An example of user output generation is given in section 7.3.

List 3: The default USROUT subroutine

```
1. C=====
2. C
3. C *** ISORROPIA CODE
4. C *** SUBROUTINE USROUT
5. C *** THIS SUBROUTINE IS USED TO GENERATE CUSTOMIZED OUTPUT
6. C
7. C *** COPYRIGHT 1996-2006, UNIVERSITY OF MIAMI, CARNEGIE MELLON UNIVERSITY,
8. C *** GEORGIA INSTITUTE OF TECHNOLOGY
9. C *** WRITTEN BY ATHANASIOS NENES
10. C *** http://nenes.eas.gatech.edu/ISORROPIA
11. C
12. C=====
13. C
14.     SUBROUTINE USROUT (ICODE)
15.     INCLUDE 'isrpio.inc'
16.     INCLUDE 'main.inc'
17. C
18. C INITIAL OUTPUT
19.     IF (ICODE.EQ.1) THEN
20. C
21. C OUTPUT AFTER EACH RUN
22.     ELSE IF (ICODE.EQ.2) THEN
23. C
24. C FINAL OUTPUT
25.     ELSE IF (ICODE.EQ.3) THEN
26.     ENDF
27. C
28.     RETURN
29.     END
```

3. ISORROPIA PROGRAMMING REFERENCE

Equilibrium calculations are performed by calling SUBROUTINE ISORROPIA. This is mainly the routine in which a programmer will CALL (and most of the time, the only one he has to deal with). In addition, ISORROPIA v1.7 offers a set of utility subroutines which allows the modeler to configure, obtain information and access the error system of ISORROPIA. The routines are versatile in the sense that they can be called at any time, regardless if a run is going to be performed or not. Running ISORROPIA is simplified, since the CALL command has relatively few arguments. In addition, if ISORROPIA is embedded into a larger model, configuration and error access can be done from host model's routines. For example, configuration can be done once, in the initialization module of an airshed model, or the error system can be accessed from an airshed's error-handling routine. The interface routines and their calling syntax follows.

3.1 SUBROUTINE ISORROPIA

This is the routine that runs the ISORROPIA (v 1.7) model.

Calling Syntax

```
CALL ISORROPIA (WI, RHI, TEMPI, CNTRL, WT, GAS, AERLIQ, AERSLD,  
              SCASI, OTHER)
```

Input Variables

- **WI**, DOUBLE PRECISION array of length [5]. Concentration of precursors, expressed in moles m⁻³ air. Depending on the type of problem solved, WI contains either *gas+aerosol* or *aerosol only* concentrations.

Each array element contains:

WI (1)	sodium
WI (2)	sulfate
WI (3)	ammonium
WI (4)	nitrate
WI (5)	chloride

- **RHI**, DOUBLE PRECISION variable. Ambient relative humidity expressed on a (0,1) scale.
- **TEMPI**, DOUBLE PRECISION variable. Ambient temperature expressed in Kelvins.
- **CNTRL**, DOUBLE PRECISION array of length [2]. Characterizes the problem solved.

CNTRL (1)	Defines the type of problem solved. 0 - Forward problem is solved. In this case, array WI contains GAS and AEROSOL concentrations together. 1 - Reverse problem is solved. In this case, array WI contains AEROSOL concentrations only.
-------------	---

CNTRL (2)	Defines the aerosol state 0 - The aerosol can have both solid+liquid phases (deliquescent). 1 - The aerosol is in only liquid state (metastable aerosol)
-------------	--

Output Variables

- **WT**, DOUBLE PRECISION array of length [05], total concentrations (gas+aerosol) of species, expressed in moles m^{-3} air. If the forward problem is solved ($\text{CNTRL}(1)=0$), array WT is identical to input array WI. Each array element contains:

WT (1)	total sodium
WT (2)	total sulfate
WT (3)	total ammonium
WT (4)	total nitrate
WT (5)	total chloride

CHAPTER 3: ISORROPIA PROGRAMMING REFERENCE

- **GAS**, DOUBLE PRECISION array of length [03]. Gaseous species concentrations, expressed in moles m^{-3} air. Each array element contains:

GAS (1) NH_3
GAS (2) HNO_3
GAS (3) HCl

- **AERLIQ**, DOUBLE PRECISION array of length [12]. Liquid aerosol species concentrations, expressed in moles m^{-3} air. Each array element contains:

AERLIQ (01) $\text{H}^+_{(\text{aq})}$
AERLIQ (02) $\text{Na}^+_{(\text{aq})}$
AERLIQ (03) $\text{NH}_4^+_{(\text{aq})}$
AERLIQ (04) $\text{Cl}^-_{(\text{aq})}$
AERLIQ (05) $\text{SO}_4^{-2}_{(\text{aq})}$
AERLIQ (06) $\text{HSO}_4^-_{(\text{aq})}$
AERLIQ (07) $\text{NO}_3^-_{(\text{aq})}$
AERLIQ (08) H_2O
AERLIQ (09) $\text{NH}_{3(\text{aq})}$ (undissociated)
AERLIQ (10) $\text{HCl}_{(\text{aq})}$ (undissociated)
AERLIQ (11) $\text{HNO}_{3(\text{aq})}$ (undissociated)
AERLIQ (12) $\text{OH}^-_{(\text{aq})}$

- **AERSLD**, DOUBLE PRECISION array of length [09]. Solid aerosol species concentrations, expressed in moles m^{-3} air. Each array element contains:

AERSLD (01) $\text{NaNO}_{3(\text{s})}$
AERSLD (02) $\text{NH}_4\text{NO}_{3(\text{s})}$
AERSLD (03) $\text{NaCl}_{(\text{s})}$
AERSLD (04) $\text{NH}_4\text{Cl}_{(\text{s})}$
AERSLD (05) $\text{Na}_2\text{SO}_{4(\text{s})}$
AERSLD (06) $(\text{NH}_4)_2\text{SO}_{4(\text{s})}$
AERSLD (07) $\text{NaHSO}_{4(\text{s})}$
AERSLD (08) $\text{NH}_4\text{HSO}_{4(\text{s})}$
AERSLD (09) $(\text{NH}_4)_4\text{H}(\text{SO}_4)_2_{(\text{s})}$

- **IACALCI**, INTEGER variable. Method of activity coefficient calculation:
 - 0 Calculate coefficients during runtime.
 - 1 Use precalculated tables.The default value of **IACALCI** is 1.
- **EPSI**, DOUBLE PRECISION variable. Defines the convergence criterion for all iterative processes in ISORROPIA, except those for activity coefficient calculations (**EPSACTI** controls that). The default value of **EPSI** is 10^{-6} .
- **MAXIT**, INTEGER variable. Defines the maximum number of iterations for all iterative processes in ISORROPIA, except those for activity coefficient calculations (**NSWEEPI** controls that). The default value of **MAXIT** is 100.
- **NSWEEPI**, INTEGER variable. Defines the maximum number of iterations for activity coefficient calculations. The default value of **NSWEEPI** is 4.
- **EPSACTI**, DOUBLE PRECISION variable. Defines the convergence criterion for activity coefficient calculations. The default value of **EPSACTI** is 5×10^{-2} .
- **NDIV**, INTEGER variable. Defines the number of subdivisions for root tracking, when the bisection method is used for numerically solving equilibrium equations. Normally this parameter should not have to change, but is included for completeness. The default value of **NDIV** is 5.
- **NADJI**, INTEGER variable. Defines whether ISORROPIA will force conservation of mass up to the precision of the machine.
 - 0 Normal calculation mode.
 - 1 After calculating the thermodynamic solution, “adjust” concentrations of species so that total conservation of mass is satisfied to machine precision (i.e., 10^{-14}).The default value of **NADJI** is 0.

NOTE: If negative values are given for a parameter, it is ignored and the current value is used instead.

Output Variables: None

3.3 SUBROUTINE GETPARAM

This subroutine obtains the current values of solution parameters.

Calling Syntax

```
CALL GETPARAM (WFTYPEPI, IACALCI, EPSI, MAXITI, NSWEEPI, EPSACTI,  
              NDIVI, NADJI)
```

Input Variables: None

Output Variables:

- **WFTYPEPI**, INTEGER variable. Defines the type of weighting algorithm for the solution in Mutual Deliquescence Regions (MDR's):
 - 0 MDR's are assumed dry. This is equivalent to the approach used by SEQUILIB.
 - 1 The solution is assumed "half" dry and "half" wet throughout the MDR.
 - 2 The solution is a relative-humidity weighted mean of the dry and wet solutions (as defined in Nenes *et al.*, 1998a)

- **IACALCI**, INTEGER variable. Method of activity coefficient calculation:
 - 0 Calculate coefficients during runtime.
 - 1 Use precalculated tables.

- **EPSI**, DOUBLE PRECISION variable. Defines the convergence criterion for all iterative processes in ISORROPIA, except those for activity coefficient calculations (**EPSACTI** controls that).

- **MAXIT**, INTEGER variable. Defines the maximum number of iterations for all iterative processes in ISORROPIA, except those for activity coefficient calculations (NSWEEPI controls that).
- **NSWEEPI**, INTEGER variable. Defines the maximum number of iterations for activity coefficient calculations.
- **EPSACTI**, DOUBLE PRECISION variable. Defines the convergence criterion for activity coefficient calculations.
- **NDIV**, INTEGER variable. Defines the number of subdivisions for root tracking, when the bisection method is used for numerically solving equilibrium equations. Normally this parameter should not have to change, but is included for completeness.
- **NADJI**, INTEGER variable. Defines whether ISORROPIA will force conservation of mass up to the precision of the machine.
 - 0 Normal calculation mode.
 - 1 After calculating the thermodynamic solution, “adjust” concentrations of species so that total conservation of mass is satisfied to machine precision (i.e., 10^{-14}).

3.4 SUBROUTINE ISERRINF

This is the routine that accesses the error system and obtains a copy of the error stack. To print error messages, use SUBROUTINE ERRSTAT.

Calling Syntax

```
CALL ISERRINF (ERRSTKI, ERRMSGI, NOFERI, STKOFLL)
```

Input Variables: None

Output Variables:

- **ERRSTKI**, INTEGER array of length [25]. This is the error stack, with the numerical codes of errors that may have occurred in a run, in reverse order of appearance. The size of the stack varies, but the user can obtain the exact length of it from SUBROUTINE ISORINF (parameter NERR).
- **ERRMSGI**, CHARACTER*40 array of length [25]. Contains information related to the corresponding element of array ERRSTKI. This information is used when SUBROUTINE ERRSTAT is used to printout error messages. The size of this array varies, but the user can obtain the exact length of it from SUBROUTINE ISORINF (parameter NERR).
- **NOFERI**, INTEGER variable. The number of errors (warning and fatal) that occurred in the most recent run. If no errors were detected, then NOFERI=0.
- **STKOFI**, LOGICAL variable. Stack overflow flag. When STKOFI=.TRUE., then more errors appeared than the stack could hold, and were not saved. Otherwise, STKOFI=.FALSE.

NOTE: If a fatal error has occurred, it is only one (fatal errors interrupt calculation) and it is the last error found in the error stack, i.e. the error code is found in ERRSTKI (NOFERI).

3.5 SUBROUTINE ERRSTAT

This routine is used for printing error messages.

Calling Syntax

```
CALL ERRSTAT ( IO , IERR , ERRINF )
```

Input Variables

- **IO**, INTEGER variable. This is the unit number where the error message is going to be written to. If output is intended to be saved in a file, unit “IO” has to previously be connected via a FORTRAN OPEN statement.
- **IERR**, INTEGER variable. This is the error code of the error message printed. It is obtained from the error stack (look at paragraph 3.4 for more details).
- **ERRINF**, CHARACTER variable (of any length). The information related to the error code IERR. This information is obtained from the error stack (look at paragraph 3.4 for more details).

Output Variables: None

NOTE: An example of using SUBROUTINE ERRSTAT can be found in paragraph 8.1.

3.6 SUBROUTINE ISORINF

This routine obtains information about ISORROPIA (ex., version number and storage requirements).

Calling Syntax

```
SUBROUTINE ISORINF (VERSI, NCMP, NION, NAQGAS, NSOL, NERR, TIN,  
                  GRT)
```

Input Variables: None

Output Variables

- **VERSI**, CHARACTER*14 variable. Version/date information of ISORROPIA code.
- **NCMP**, INTEGER variable. The number of components in input array WI.

CHAPTER 3: ISORROPIA PROGRAMMING REFERENCE

- **NION**, INTEGER variable. The maximum number of ionic species in the aqueous phase.
- **NAQGAS**, INTEGER variable. The maximum number of undissociated species in the aqueous phase.
- **NSOL**, INTEGER variable. The maximum number of solids in the aerosol phase.
- **NERR**, INTEGER variable. The size of the error stack (maximum number of errors that can be traced in a run).
- **TIN**, DOUBLE PRECISION variable. The value used for a very small number.
- **GRT**, DOUBLE PRECISION variable. The value used for a very large number.

4. ISORROPIA STRUCTURE

This section describes the internal structure and calling sequence of the subroutines included in ISORROPIA. The main calling tree is shown in Figure 2, while the calling sequence for each “submodel” is shown in Figures 3 to 8. A more detailed analysis of each subcase is given in Chapter 5. The thermodynamic equations, together with mass conservation and electroneutrality are solved by a “CALC” subroutine that corresponds to the input conditions and problem type (forward or reverse).

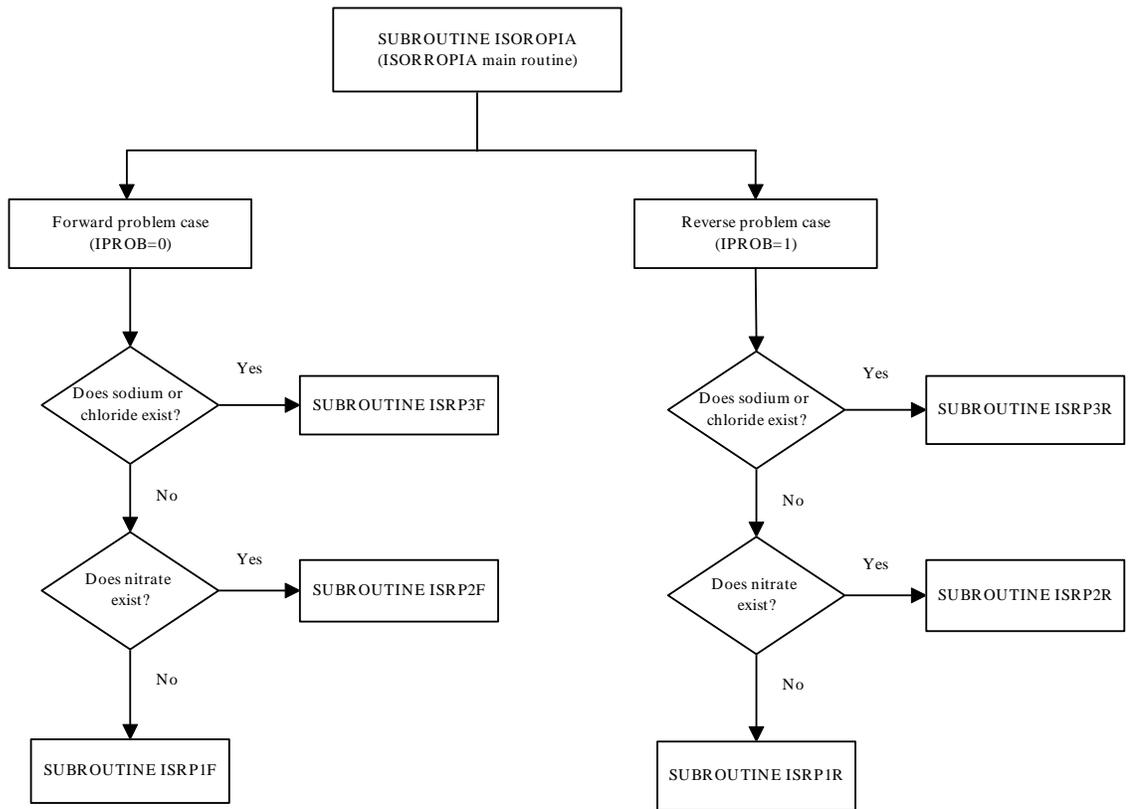


Figure 2: The main calltree of ISORROPIA, with the six basic “submodels”.

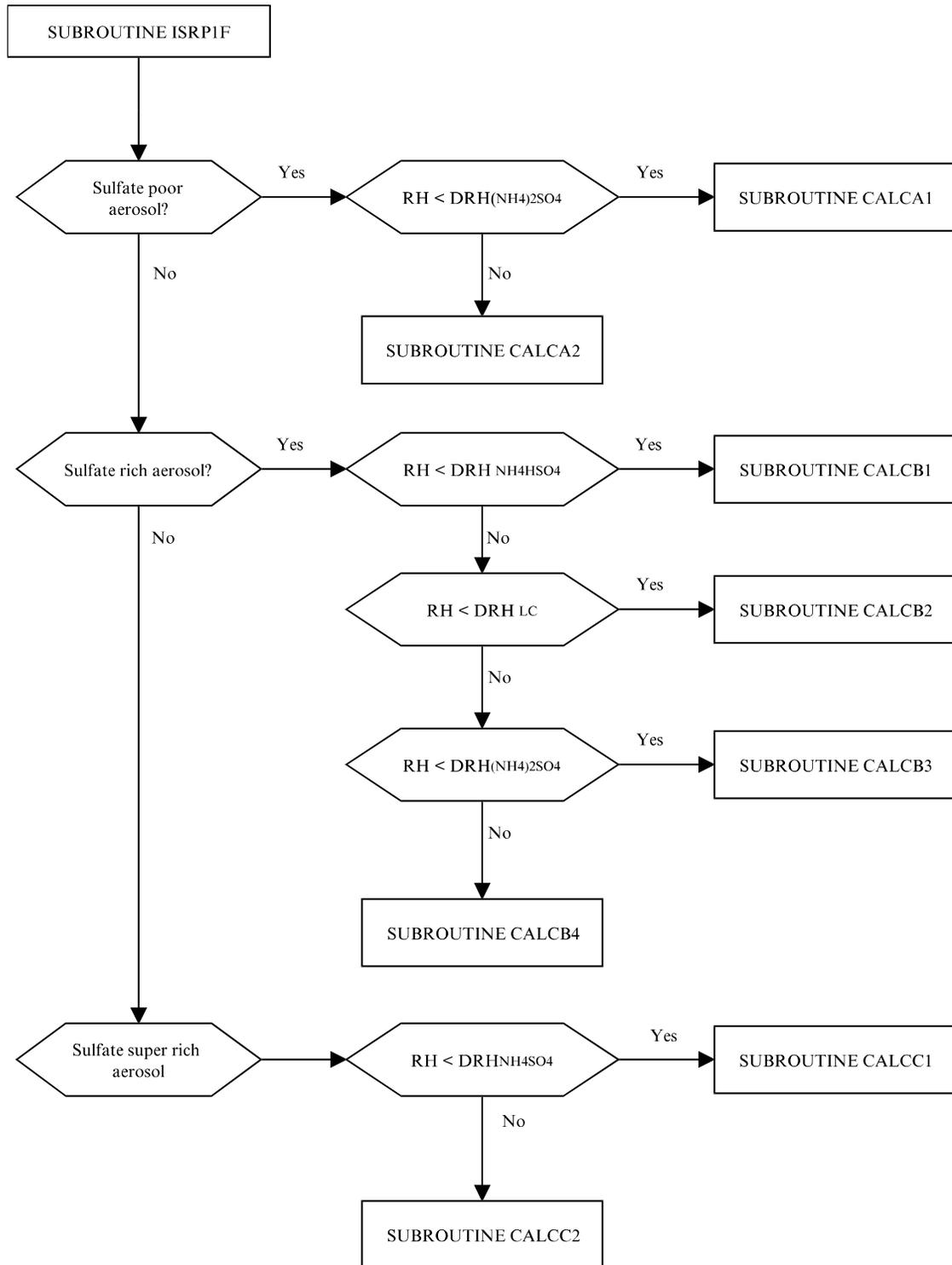


Figure 3: The calltree for solving the forward problem of an ammonium-sulfate aerosol.

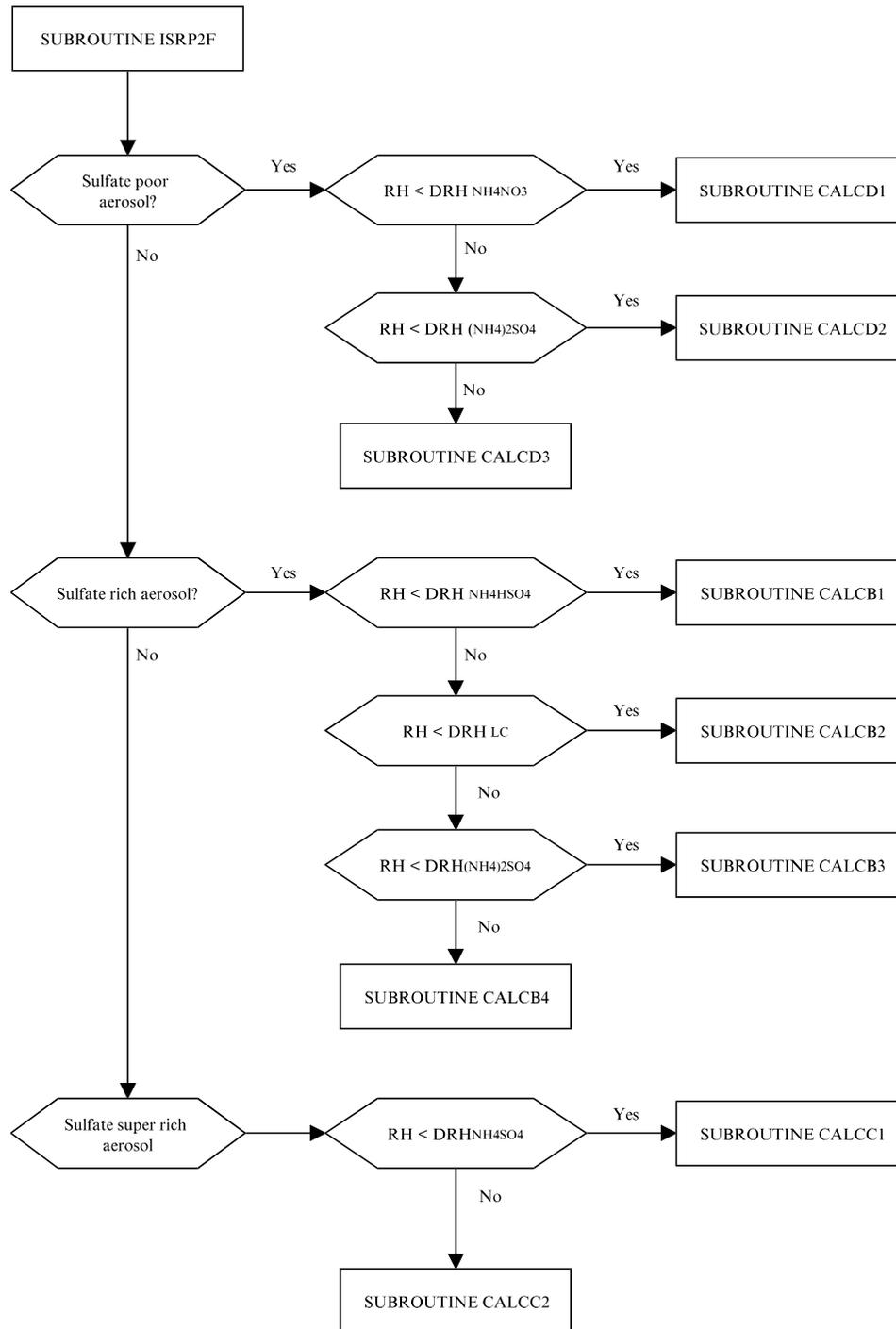


Figure 4: The calltree for solving the forward problem of an ammonium-sulfate-nitrate aerosol.

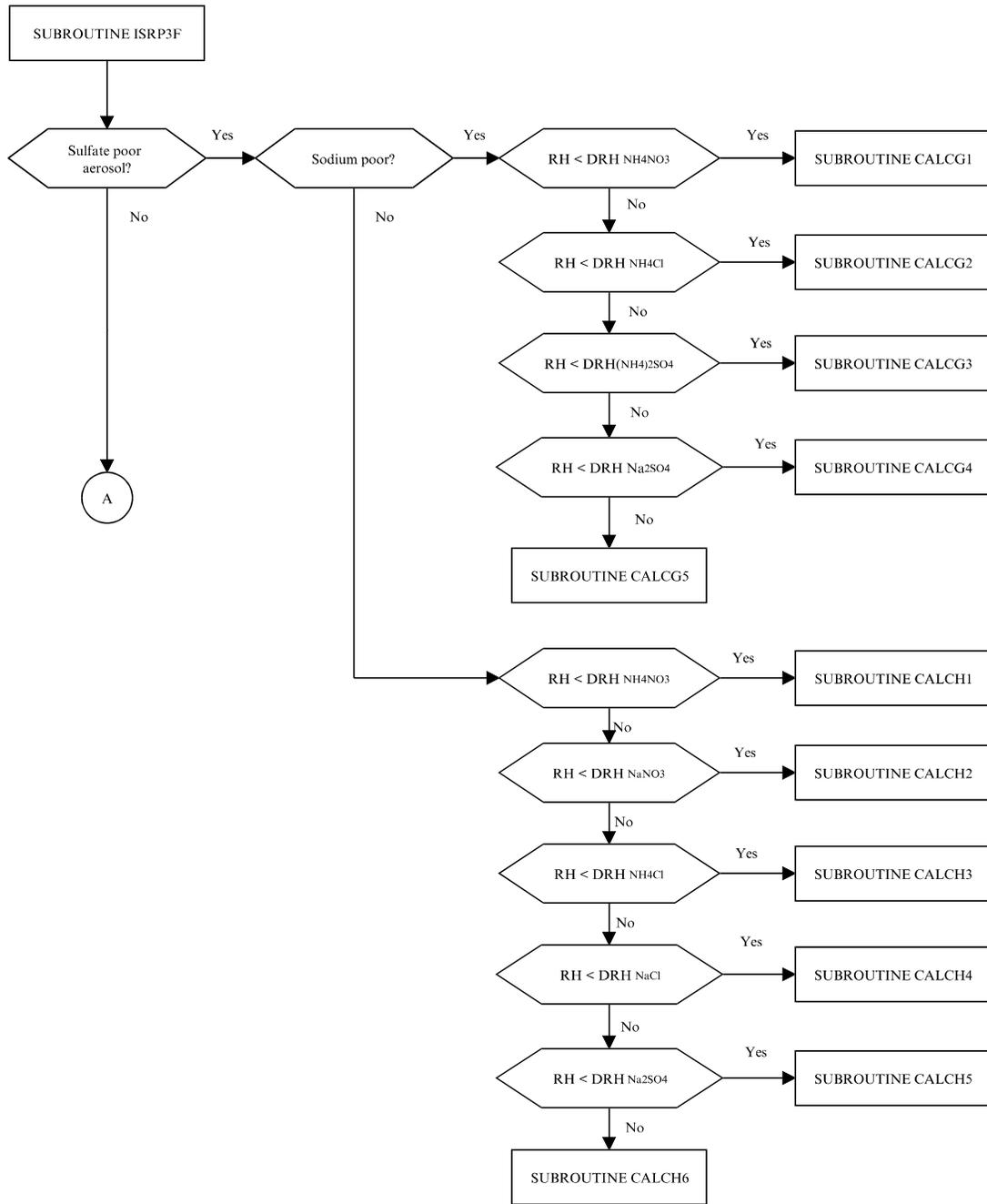


Figure 5 The calltree for solving the forward problem of a sodium-ammonium-sulfate-nitrate-chloride aerosol.

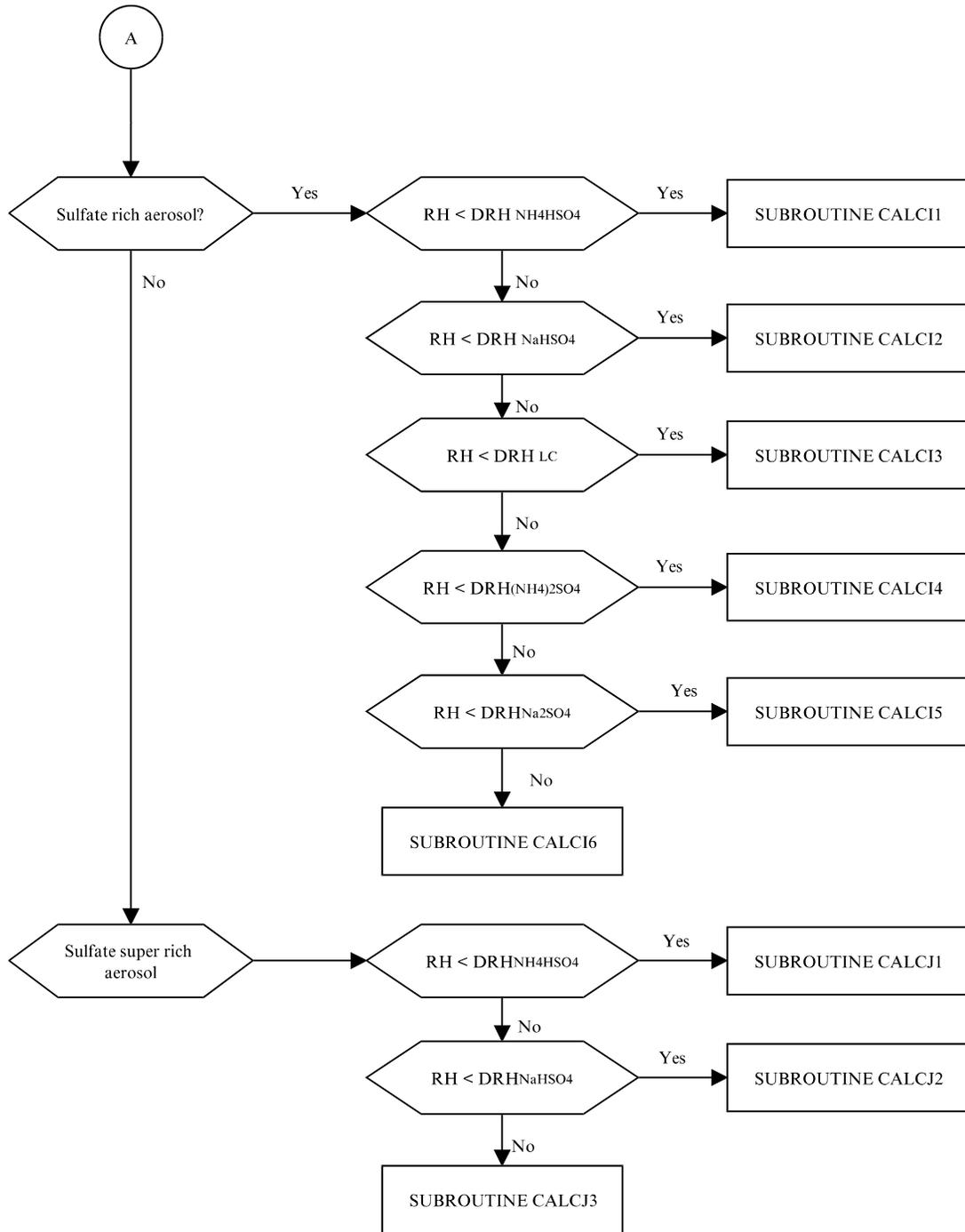


Figure 5 (Continued)

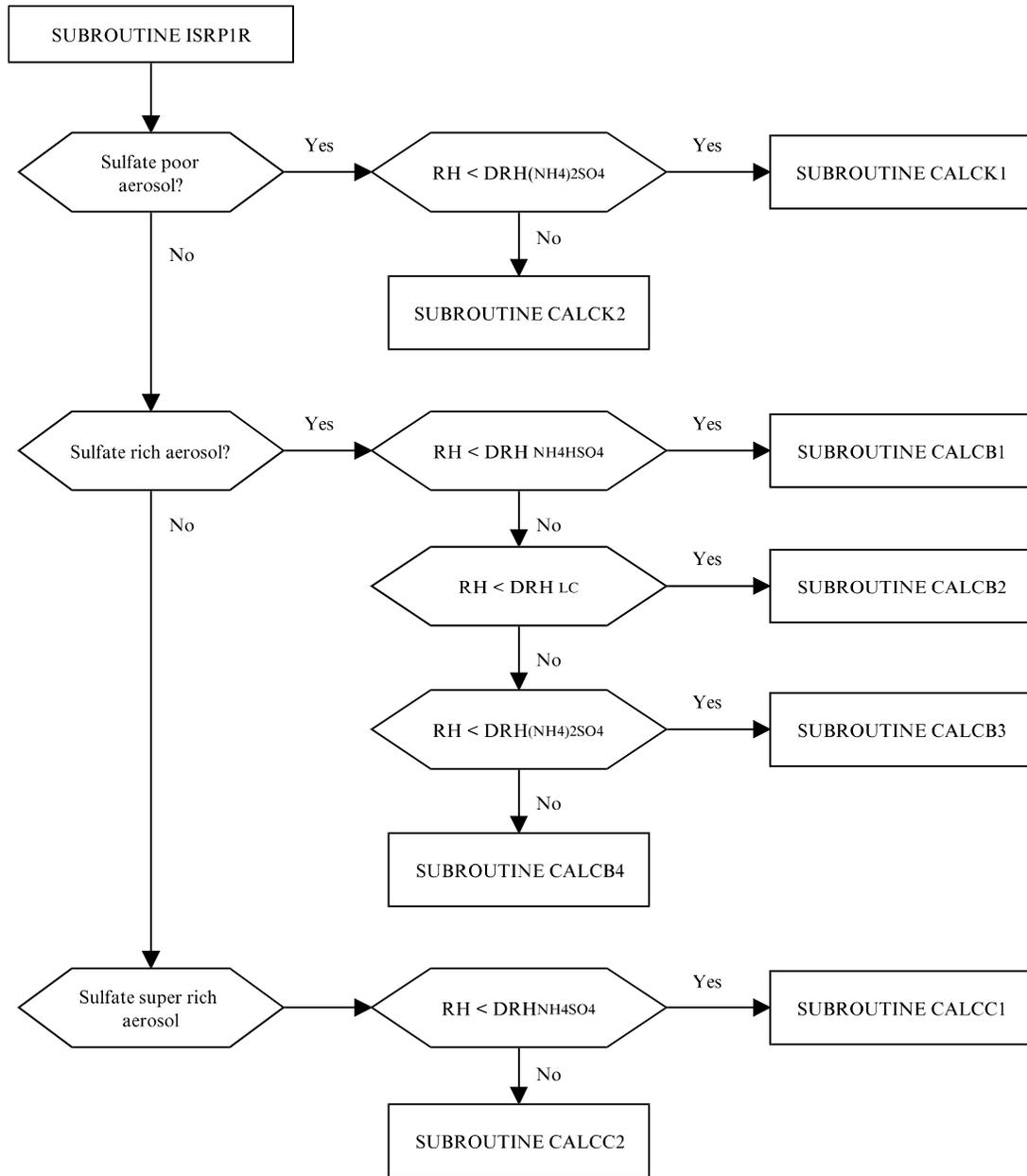


Figure 6: The calltree for solving the reverse problem of an ammonium-sulfate aerosol.

CHAPTER 4: ISORROPIA STRUCTURE

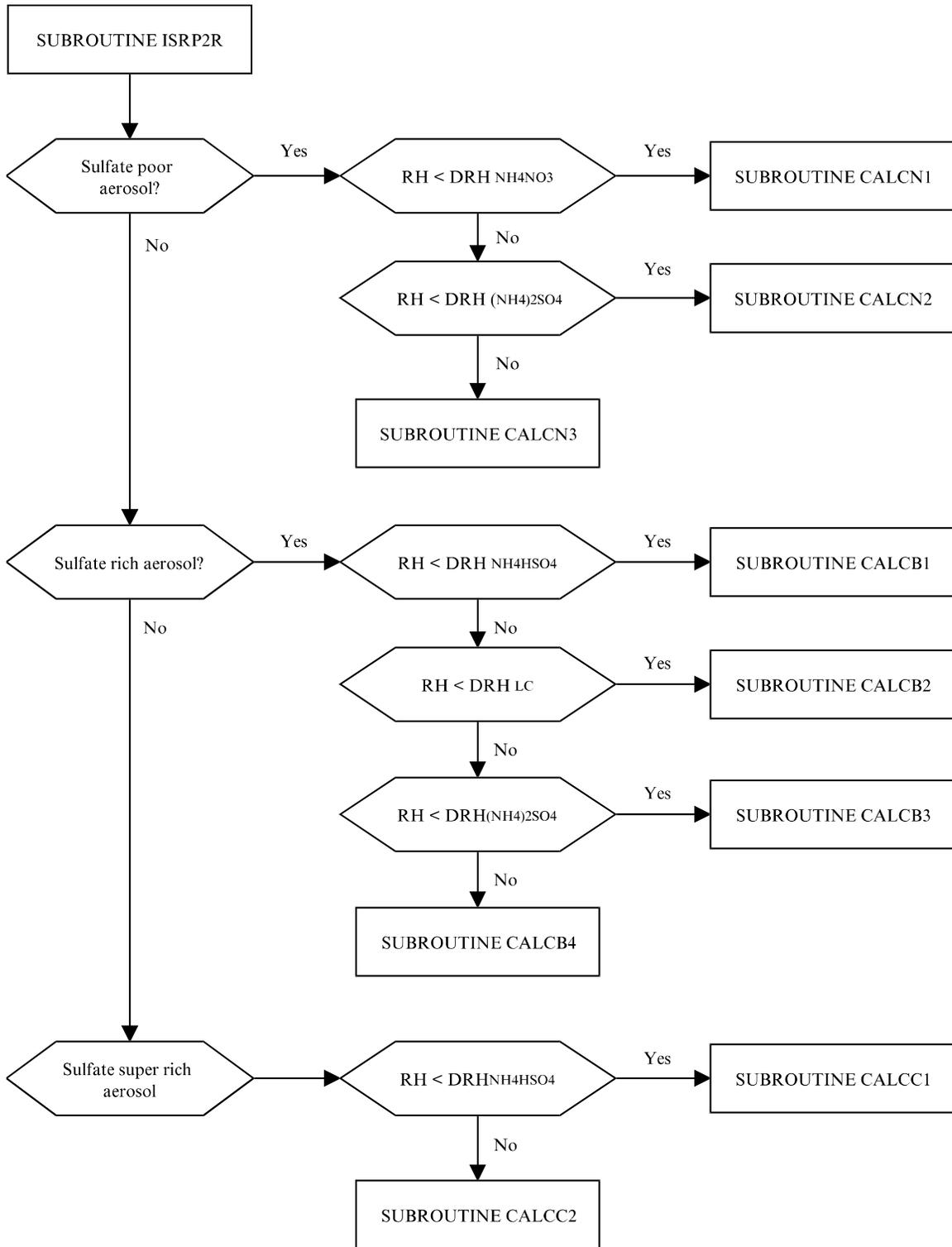


Figure 7: The calltree for solving the reverse problem of an ammonium-sulfate-nitrate aerosol.

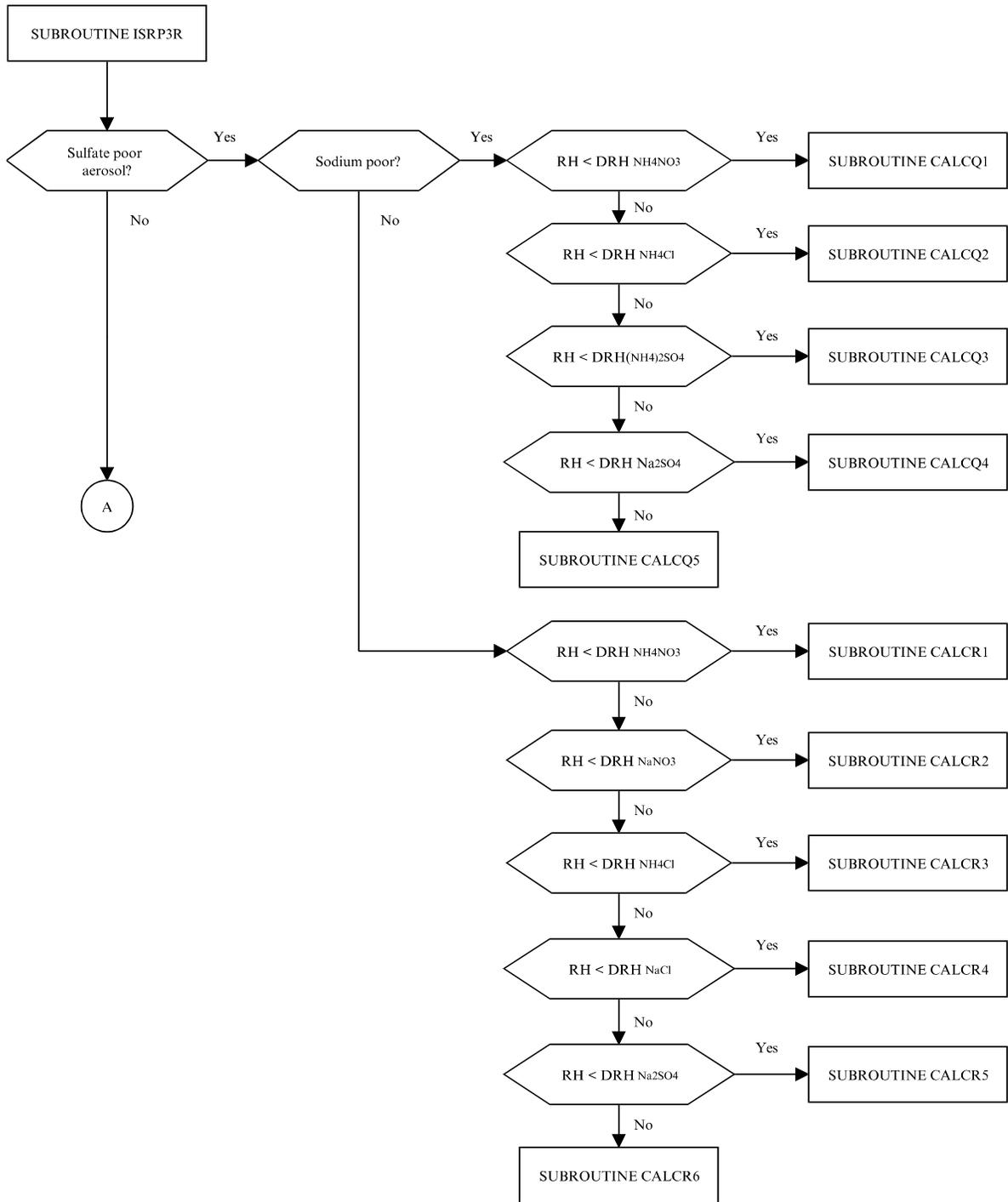


Figure 8: The calltree for solving the reverse problem of a sodium-ammonium-sulfate-nitrate-chloride aerosol.

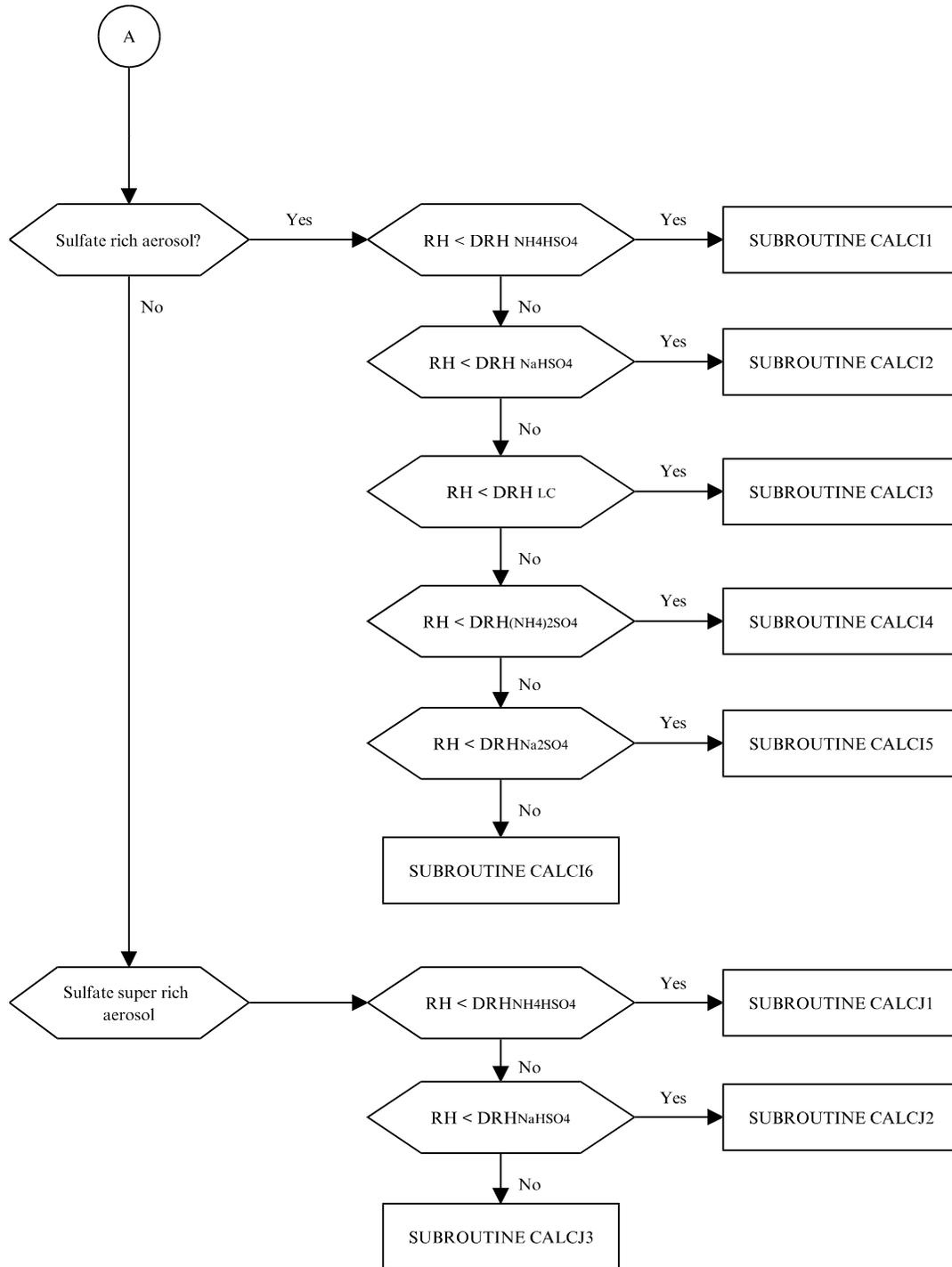


Figure 8: (Continued)

5. SUBROUTINE REFERENCE

This section briefly describes most of the subroutines used in ISORROPIA. In the following routines, the term “objective function” refers to a function which is numerically solved for its root. The reader can refer to Nenes *et.al* (1998a) for a more detailed description of any terms.

SUBROUTINE ISORROPIA, SETPARM, GETPARM, ISERRINF, ERRSTAT, ISORINF

These are the interface routines of ISORROPIA (v.1.1 and above). A detailed description of them can be found in Chapter 3.

SUBROUTINE ISOPLUS

This is the interface routine of ISORROPIA v.1.0, and is included for backward compatibility with models that use the calling conventions of this older version. This routine is described in detail in appendix B.

BLOCK DATA BLKISO

This subroutine provides initial (default) values to program parameters via data statements.

SUBROUTINE INIT1

This subroutine initializes the global variables for aerosol systems composed only of ammonium and sulfate.

CHAPTER 5: SUBROUTINE REFERENCE

SUBROUTINE INIT2

This subroutine initializes the global variables for aerosol systems composed only of ammonium, nitrate and sulfate.

SUBROUTINE INIT3

This subroutine initializes the global variables for aerosol systems composed of ammonium, sodium, chloride, nitrate and sulfate.

SUBROUTINE ISRP1F

This routine handles the forward problem for ammonium-sulfate aerosol systems. The possible species are determined from the sulfate ratio and ambient relative humidity. Cases solved by subroutines CALCA1, CALCA2, CALCB1 through CALCB4, CALCC1 and CALCC2 are called here. The calling sequence is shown in Figure 3.

SUBROUTINE ISRP2F

This routine handles the forward problem for ammonium-nitrate-sulfate aerosol systems. The possible species are determined from the sulfate ratio and ambient relative humidity. Cases solved by subroutines CALCD1 through CALCD3, CALCB1 through CALCB4, CALCC1 and CALCC2 are called here. The calling sequence is shown in Figure 4.

SUBROUTINE ISRP3F

This routine handles the forward problem for sodium-ammonium-nitrate-sulfate-chloride aerosol systems. The possible species are determined from the sulfate ratio, the sodium ratio and from the ambient relative humidity. Cases solved by subroutines CALCG1 through CALCG5, CALCH1

CHAPTER 5: SUBROUTINE REFERENCE

through CALCH6, CALCI1 through CALCI6, CALCJ1 through CALCJ3 are called here. The calling sequence is shown in Figure 5.

SUBROUTINE ISRP1R

This routine handles the reverse problem for ammonium-sulfate aerosol systems. The possible species are determined from the sulfate ratio and ambient relative humidity. Cases solved by subroutines CALCK1, CALCK2, CALCB1 through CALCB4, CALCC1 and CALCC2 are called here. Figure 6 displays the calling sequence.

SUBROUTINE ISRP2R

This subroutine is the driver routine for the reverse problem of an ammonium-nitrate-sulfate aerosol system. The possible species are determined from the sulfate ratio and ambient relative humidity. Cases solved by subroutines CALCN1 through CALCN3, CALCB1 through CALCB4, CALCC1 and CALCC2 are called here. The calling sequence is shown in Figure 7.

SUBROUTINE ISRP3R

This routine handles the reverse problem for sodium-ammonium-nitrate-sulfate-chloride aerosol systems. The possible species are determined from the sulfate ratio, the sodium ratio and from the ambient relative humidity. Cases solved by subroutines CALCQ1 through CALCQ5, CALCR1 through CALCR6, CALCI1 through CALCI6, CALCJ1 through CALCJ3 are called here. The calling sequence is shown in Figure 8.

CHAPTER 5: SUBROUTINE REFERENCE

SUBROUTINE CALCACT

This subroutine calculates the activity coefficients in the aqueous phase using Bromley's method. Binary (single component) activity coefficients are calculated by the Kusik-Meissner relation, either during runtime (SUBROUTINE KMFUL) or obtained from precalculated tables (SUBROUTINE KMTAB).

SUBROUTINE ADJUST

This subroutine adjusts to satisfy mass balance up to machine precision. It calculates the excess of each aerosol precursor, and if it exists, it adjusts sequentially aerosol phase species which contain the excess precursor.

SUBROUTINE CALCA1

This subroutine solves the forward problem of a sulfate poor, ammonium-sulfate aerosol system, when the relative humidity is below the DRH of $(\text{NH}_4)_2\text{SO}_4$. The following species are possible:

- Solids: $(\text{NH}_4)_2\text{SO}_4$
- Aqueous phase: Not present

Solid $(\text{NH}_4)_2\text{SO}_4$ is calculated from the amount of sulfates available. Excess ammonia is partitioned between gas and aqueous phase according to the $\text{NH}_{3(g)} \longleftrightarrow \text{NH}_{4(aq)}^+ + \text{OH}_{(aq)}^-$ equilibrium.

SUBROUTINE CALCA2

This subroutine solves the forward problem of a sulfate poor, ammonium-sulfate aerosol system, when the relative humidity is above the DRH of $(\text{NH}_4)_2\text{SO}_4$. The following species are possible:

CHAPTER 5: SUBROUTINE REFERENCE

- Solids: None
- Aqueous phase: Present

FUNCTION FUNCA2 calculates the concentration of ions from the $NH_{3(g)} \longleftrightarrow NH_{4(aq)}^+ + OH_{(aq)}^-$ equilibrium, based on an estimation of H^+ . This process is iterated with respect to the concentration of H^+ , until electroneutrality is satisfied.

SUBROUTINE CALCB1

This subroutine solves the forward problem of a sulfate rich, ammonium-sulfate aerosol system, when the relative humidity is below the DRH of NH_4HSO_4 . The following species are possible:

- Solids: $(NH_4)_3H(SO_4)_2$, $(NH_4)_2SO_4$, NH_4HSO_4
- Aqueous phase: Present when $RH \geq MDRH$, not present when $RH < MDRH$

SUBROUTINE CALCB1A

This subroutine solves the forward problem of a sulfate rich, ammonium-sulfate aerosol system, when the relative humidity is below the MDRH regions of SUBROUTINE CALCB1. The following species are possible:

- Solids: $(NH_4)_3H(SO_4)_2$, NH_4HSO_4 or $(NH_4)_3H(SO_4)_2$, $(NH_4)_2SO_4$
- Aqueous phase: Not present

A material balance is performed which calculates the amount of $(NH_4)_3H(SO_4)_2$, $(NH_4)_2SO_4$ and NH_4HSO_4 . Only two of the three salts can exist simultaneously, one of which is always $(NH_4)_3H(SO_4)_2$.

SUBROUTINE CALCB1B

CHAPTER 5: SUBROUTINE REFERENCE

This subroutine solves the forward problem of a sulfate rich, ammonium-sulfate aerosol system, when the relative humidity is in the MDRH regions of SUBROUTINE CALCB1. The following species are possible:

- Solids: $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$, NH_4HSO_4 or $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$, $(\text{NH}_4)_2\text{SO}_4$
- Aqueous phase: Present

The solution is assumed to be the weighted average of two states; one being the dry aerosol calculated by SUBROUTINE CALCB1A, and the other, a liquid phase calculated by SUBROUTINE CALCB2.

SUBROUTINE CALCB2

This subroutine solves the forward problem of a sulfate rich, ammonium-sulfate aerosol system, when the relative humidity is between the DRH of NH_4HSO_4 and $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$. The following species are possible:

- Solids: $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$, $(\text{NH}_4)_2\text{SO}_4$
- Aqueous phase: Present if in an MDRH, or if NH_4HSO_4 is possible. Otherwise, no aqueous phase is present.

There are two subcases possible, depending on the type of salts that can exist.

- When both $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$ and $(\text{NH}_4)_2\text{SO}_4$ are possible, SUBROUTINE CALCB2A is called.
- When only $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$ is possible, SUBROUTINE CALCB2B is called.

SUBROUTINE CALCB2A1

CHAPTER 5: SUBROUTINE REFERENCE

This subroutine solves the forward problem of a sulfate rich, ammonium-sulfate aerosol system, when the relative humidity is between the DRH of NH_4HSO_4 and $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$. The following species are possible:

- Solids: $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$, $(\text{NH}_4)_2\text{SO}_4$
- Aqueous phase: Present when $\text{RH} \geq \text{MDRH}$, not present when $\text{RH} < \text{MDRH}$

SUBROUTINE CALCB2A2

This subroutine solves the forward problem of a sulfate rich, ammonium-sulfate aerosol system, when the relative humidity is above the DRH of NH_4HSO_4 and in the MDRH regions of SUBROUTINE CALCB2A. The following species are possible:

- Solids: $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$, $(\text{NH}_4)_2\text{SO}_4$
- Aqueous phase: Present

The solution is assumed to be the weighted average of two states; one being the dry aerosol calculated by SUBROUTINE CALCB2A1, and the other, a liquid phase calculated by SUBROUTINE CALCB3.

SUBROUTINE CALCB2B

This subroutine solves the forward problem of a sulfate rich, ammonium-sulfate aerosol system, when the relative humidity is between the DRH of NH_4HSO_4 and $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$. The following species are possible:

- Solids: $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$

CHAPTER 5: SUBROUTINE REFERENCE

- Aqueous phase: Present

The system is solved iteratively, using the amount of dissolved $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$ as the estimation variable. For each estimation, FUNCTION `FUNCB2A` calculates the ionic equilibria in the aqueous phase. The solubility product of $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$ is used as the objective function.

SUBROUTINE CALCB3

This subroutine solves the forward problem of a sulfate rich, ammonium-sulfate aerosol system, when the relative humidity lies between the DRH of $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$ and $(\text{NH}_4)_2\text{SO}_4$. The following species are possible:

- Solids: $(\text{NH}_4)_2\text{SO}_4$
- Aqueous phase: Present

There are two possible cases, depending on if solids can form. When solid $(\text{NH}_4)_2\text{SO}_4$ is possible (i.e. when the sulfate ratio is greater than 1.5), SUBROUTINE `CALCB3A` is called, otherwise, SUBROUTINE `CALCB3B` is called.

SUBROUTINE CALCB3A

This subroutine solves the forward problem of a sulfate rich, ammonium-sulfate aerosol system, when the relative humidity lies between the DRH of $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$ and $(\text{NH}_4)_2\text{SO}_4$. The following species are possible:

CHAPTER 5: SUBROUTINE REFERENCE

- Solids: $(\text{NH}_4)_2\text{SO}_4$
- Aqueous phase: Present

The system is solved iteratively, using the amount of dissolved $(\text{NH}_4)_2\text{SO}_4$ dissolved as the estimation variable. For each estimation, FUNCTION FUNCB3A calculates the ionic equilibria in the aqueous phase. The solubility product of $(\text{NH}_4)_2\text{SO}_4$ is used as the objective function.

SUBROUTINE CALCB3B

This subroutine solves the forward problem of a sulfate rich, ammonium-sulfate aerosol system, when the relative humidity lies between the DRH of $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$ and $(\text{NH}_4)_2\text{SO}_4$. The following species are possible:

- Solids: None
- Aqueous phase: Present

The speciation is controlled mainly by the $\text{HSO}_4^- \rightleftharpoons \text{H}^+ + \text{SO}_4^{2-}$ equilibrium.

SUBROUTINE CALCB4

This subroutine solves the forward problem of a sulfate rich, ammonium-sulfate aerosol system, when the relative humidity is above the DRH of $(\text{NH}_4)_2\text{SO}_4$. The following species are possible:

- Solids: None
- Aqueous phase: Present

FUNCTION FUNCB4 calculates the concentration of ions from the $\text{NH}_3 \rightleftharpoons \text{NH}_4^+ + \text{OH}^-$ equilibrium, based on an estimation of H^+ . This process is iterated with respect to the concentration of H^+ , until electroneutrality is satisfied.

CHAPTER 5: SUBROUTINE REFERENCE

SUBROUTINE CALCC1

This subroutine solves the forward problem of a sulfate super-rich, ammonium-sulfate aerosol system, when the relative humidity is below the DRH of NH_4HSO_4 . The following species are possible:

- Solids: NH_4HSO_4
- Aqueous phase: Present

The system is solved iteratively, using the amount of dissolved NH_4HSO_4 as the estimation variable. For each estimation, FUNCTION FUNCC1A calculates the ionic equilibria in the aqueous phase. The solubility product of NH_4HSO_4 is used as the objective function.

SUBROUTINE CALCC2

This subroutine solves the forward problem of a sulfate super-rich, ammonium-sulfate aerosol system, when the relative humidity is above the DRH of NH_4HSO_4 . The following species are possible:

- Solids: None
- Aqueous phase: Present

The speciation is controlled mainly by the $\text{HSO}_4^- \longleftrightarrow \text{H}^+ + \text{SO}_4^{2-}$ equilibrium.

SUBROUTINE CALCD1

This subroutine solves the forward problem of a sulfate poor, ammonium-nitrate-sulfate aerosol system, when the relative humidity is below the DRH of NH_4NO_3 . The following species are possible:

CHAPTER 5: SUBROUTINE REFERENCE

- Solids: $(\text{NH}_4)_2\text{SO}_4$, NH_4NO_3
- Aqueous phase: Present when $\text{RH} \geq \text{MDRH}$, not present when $\text{RH} < \text{MDRH}$

The case of a dry aerosol ($\text{RH} < \text{MDRH}$) is solved by SUBROUTINE CALCD1A. The composition in the mutual deliquescence region is solved by SUBROUTINE CALCMDRH.

SUBROUTINE CALCD1A

This subroutine solves the forward problem of a sulfate poor, ammonium-nitrate-sulfate aerosol system, when the relative humidity is below the MDRH point of SUBROUTINE CALCD1. The following species are possible:

- Solids: $(\text{NH}_4)_2\text{SO}_4$, NH_4NO_3
- Aqueous phase: None

The amount of solid $(\text{NH}_4)_2\text{SO}_4$ is calculated from the sulfates. Solid NH_4NO_3 is calculated from the $\text{NH}_4\text{NO}_{3(s)} \longleftrightarrow \text{NH}_{3(g)} + \text{HNO}_{3(g)}$ equilibrium.

SUBROUTINE CALCD2

This subroutine solves the forward problem of a sulfate poor, ammonium-nitrate-sulfate aerosol system, when the relative humidity is between the DRH of NH_4NO_3 and $(\text{NH}_4)_2\text{SO}_4$. The following species are possible:

- Solids: $(\text{NH}_4)_2\text{SO}_4$
- Aqueous phase: Present

CHAPTER 5: SUBROUTINE REFERENCE

The system is solved iteratively, using the amount of gaseous NH_3 as the estimation variable. Based on this estimation, FUNCTION `FUNCD2` calculates the ionic equilibria in the aqueous phase and the amount of dissolved $(\text{NH}_4)_2\text{SO}_4$. The objective function used is the departure of NH_4^+ , H^+ and $\text{NH}_3(\text{g})$ concentrations from their equilibrium values, according to the reaction



SUBROUTINE CALCD3

This subroutine solves the forward problem of a sulfate poor, ammonium-nitrate-sulfate aerosol system, when the relative humidity is above the DRH of $(\text{NH}_4)_2\text{SO}_4$. The following species are possible:

- Solids: None
- Aqueous phase: Present

The system is solved iteratively, using the amount of gaseous NH_3 as the estimation variable. Based on this estimation, FUNCTION `FUNCD3` calculates the ionic equilibria in the aqueous phase. The objective function used is the departure of NH_4^+ , H^+ and $\text{NH}_3(\text{g})$ from the equilibrium state according to the reaction $\text{NH}_{3(\text{g})} + \text{H}_2\text{O}_{(\text{aq})} \longleftrightarrow \text{NH}_{4(\text{aq})}^+ + \text{OH}_{(\text{aq})}^-$.

SUBROUTINE CALCG1

This subroutine solves the forward problem of a sulfate poor, sodium poor, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is below the DRH of NH_4NO_3 .

The following species are possible:

- Solids: $(\text{NH}_4)_2\text{SO}_4$, NH_4NO_3 , NH_4Cl , Na_2SO_4

CHAPTER 5: SUBROUTINE REFERENCE

- Aqueous phase: Present when $RH \geq MDRH$, not present when $RH < MDRH$

The case of a dry aerosol ($RH < MDRH$) is solved by SUBROUTINE CALCG1A. The composition in the mutual deliquescence region is solved by SUBROUTINE CALCMDRH.

SUBROUTINE CALCG1A

This subroutine solves the forward problem of a sulfate poor, sodium poor, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is below the MDRH region of SUBROUTINE CALCG1. The following species are possible:

- Solids: $(NH_4)_2SO_4$, NH_4NO_3 , NH_4Cl , Na_2SO_4
- Aqueous phase: Not present

All of the sodium reacts with the sulfates to form Na_2SO_4 . The remaining sulfates react with ammonia to form $(NH_4)_2SO_4$. The excess NH_3 then reacts with HNO_3 and HCl to form NH_4NO_3 , NH_4Cl .

SUBROUTINE CALCG2

This subroutine solves the forward problem of a sulfate poor, sodium poor, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NH_4NO_3 and NH_4Cl . The following species are possible:

- Solids: $(NH_4)_2SO_4$, NH_4Cl , Na_2SO_4
- Aqueous phase: Present when there is NH_4NO_3 (that deliquesces) or when in a MDRH.

CHAPTER 5: SUBROUTINE REFERENCE

In the presence of NH_4NO_3 , the problem is solved by SUBROUTINE CALCG2A. When in a MDRH, SUBROUTINE CALCMDRH is called. Otherwise, the aerosol is dry, and SUBROUTINE CALCG1A is used to calculate the composition.

SUBROUTINE CALCG2A

This subroutine solves the forward problem of a sulfate poor, sodium poor, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NH_4NO_3 and NH_4Cl . An aqueous phase is also presumed to exist. The following species are possible:

- Solids: $(\text{NH}_4)_2\text{SO}_4$, NH_4Cl , Na_2SO_4
- Aqueous phase: Present.

The system is solved iteratively, with respect to the amount of gas phase HCl. For each estimation, FUNCTION FUNCG2A calculates the ionic equilibria in the aqueous phase, together with the amount of precipitated salts and gas phase NH_3 , HNO_3 . The objective function used is the departure of $\text{Cl}^-_{(aq)}$, $\text{NH}_4^+_{(aq)}$, $\text{HCl}_{(g)}$ and $\text{NH}_{3(g)}$ from the equilibrium state, according to the reaction $\text{NH}_{3(g)} + \text{HCl}_{(g)} \longleftrightarrow \text{NH}^+_{4(aq)} + \text{Cl}^-_{(aq)}$.

SUBROUTINE CALCG3

This subroutine solves the forward problem of a sulfate poor, sodium poor, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NH_4Cl and $(\text{NH}_4)_2\text{SO}_4$. The following species are possible:

- Solids: $(\text{NH}_4)_2\text{SO}_4$, Na_2SO_4

CHAPTER 5: SUBROUTINE REFERENCE

- Aqueous phase: Present when there is some NH_4NO_3 or NH_4Cl (which deliquesces), or when in a MDRH.

In the presence of NH_4NO_3 or NH_4Cl , the problem is solved by SUBROUTINE `CALCG3A`. When in a MDRH, SUBROUTINE `CALCMDRH` is called. Otherwise, the aerosol is dry, and SUBROUTINE `CALCG1A` is used to calculate the composition.

SUBROUTINE CALCG3A

This subroutine solves the forward problem of a sulfate poor, sodium poor, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NH_4Cl and $(\text{NH}_4)_2\text{SO}_4$. An additional presumption made is that nitrate and/or chloride exists in the aerosol phase. The following species are possible:

- Solids: $(\text{NH}_4)_2\text{SO}_4$, Na_2SO_4
- Aqueous phase: Present.

The system is solved iteratively, with respect to the amount of gas phase HCl. For each estimation, FUNCTION `FUNCG3A` calculates the ionic equilibria in the aqueous phase, together with the amount of precipitated salts and gas phase NH_3 , HNO_3 . The objective function used is the departure of $\text{Cl}^-_{(aq)}$, $\text{NH}_4^-_{(aq)}$, $\text{HCl}_{(g)}$ and $\text{NH}_{3(g)}$ from the equilibrium state, according to the reaction $\text{NH}_{3(g)} + \text{HCl}_{(g)} \longleftrightarrow \text{NH}_{4(aq)}^+ + \text{Cl}_{(aq)}^-$.

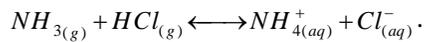
SUBROUTINE CALCG4

This subroutine solves the forward problem of a sulfate poor, sodium poor, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of $(\text{NH}_4)_2\text{SO}_4$ and Na_2SO_4 . The following species are possible:

CHAPTER 5: SUBROUTINE REFERENCE

- Solids: Na₂SO₄
- Aqueous phase: Present

The system is solved iteratively, with respect to the amount of gas phase HCl. For each estimation, FUNCTION FUNCG4 calculates the ionic equilibria in the aqueous phase, together with the amount of precipitated Na₂SO₄ and gas phase NH₃, HNO₃. The objective function used is the departure of Cl⁻_(aq), NH₄⁻_(aq), HCl_(g) and NH_{3(g)} from equilibrium, according to the reaction



SUBROUTINE CALCG5

This subroutine solves the forward problem of a sulfate poor, sodium poor, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of (NH₄)₂SO₄ and Na₂SO₄. The following species are possible:

- Solids: None
- Aqueous phase: Present

The iterative solution algorithm of SUBROUTINE CALCG4 is used, with the difference that Na₂SO₄ does not precipitate.

SUBROUTINE CALCH1

This subroutine solves the forward problem of a sulfate poor, sodium rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is below the DRH of NH₄NO₃.

The following species are possible:

- Solids: Na₂SO₄, NaNO₃, NaCl, NH₄NO₃, NH₄Cl

CHAPTER 5: SUBROUTINE REFERENCE

- Aqueous phase: Present when $RH \geq MDRH$, not present when $RH < MDRH$

The case of a dry aerosol ($RH < MDRH$) is solved by SUBROUTINE CALCH1A. The composition in the mutual deliquescence region is solved by SUBROUTINE CALCMDRH.

SUBROUTINE CALCH1A

This subroutine solves the forward problem of a sulfate poor, sodium rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is below the MDRH regions of SUBROUTINE CALCH1. The following species are possible:

- Solids: Na_2SO_4 , $NaNO_3$, $NaCl$, NH_4NO_3 , NH_4Cl
- Aqueous phase: Not present

The sodium reacts with all of the sulfates to produce Na_2SO_4 . The remaining sodium first reacts to form $NaNO_3$, then if possible, $NaCl$. HCl and HNO_3 then react with NH_3 to form NH_4NO_3 and NH_4Cl .

SUBROUTINE CALCH2

This subroutine solves the forward problem of a sulfate poor, sodium rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NH_4NO_3 and $NaNO_3$. The following species are possible:

- Solids: Na_2SO_4 , $NaNO_3$, $NaCl$, NH_4Cl
- Aqueous phase: Present when NH_4NO_3 exists or when $RH \geq MDRH$, not present when $RH < MDRH$

CHAPTER 5: SUBROUTINE REFERENCE

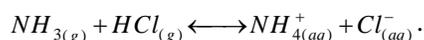
The case of a dry aerosol ($RH < MDRH$) is solved by SUBROUTINE CALCH1A. The composition in the mutual deliquescence region is solved by SUBROUTINE CALCMDRH.

SUBROUTINE CALCH2A

This subroutine solves the forward problem of a sulfate poor, sodium rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NH_4NO_3 and $NaNO_3$. An aqueous phase is also presumed to exist. The following species are possible:

- Solids: Na_2SO_4 , $NaNO_3$, $NaCl$, NH_4Cl
- Aqueous phase: Present.

The system is solved iteratively, with respect to the amount of gas phase HCl. For each estimation, FUNCTION FUNCH2A calculates the ionic equilibria in the aqueous phase, together with the amount of precipitated salts and gas phase NH_3 , HNO_3 . The objective function used is the departure of $Cl^-_{(aq)}$, $NH_4^+_{(aq)}$, $HCl_{(g)}$ and $NH_{3(g)}$ from equilibrium, according to the reaction



SUBROUTINE CALCH3

This subroutine solves the forward problem of a sulfate poor, sodium rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of $NaNO_3$ and $NaCl$. An aqueous phase is presumed to exist. The following species are possible:

- Solids: Na_2SO_4 , $NaCl$, NH_4Cl
- Aqueous phase: Present.

CHAPTER 5: SUBROUTINE REFERENCE

The iterative solution algorithm of SUBROUTINE CALCH2 is used, with the difference that NaNO_3 does not precipitate.

SUBROUTINE CALCH4

This subroutine solves the forward problem of a sulfate poor, sodium rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NaCl and NH_4Cl . The following species are possible:

- Solids: Na_2SO_4 , NH_4Cl
- Aqueous phase: Present.

The iterative solution algorithm of SUBROUTINE CALCH2 is used, with the difference that NaNO_3 and NaCl do not precipitate.

SUBROUTINE CALCH5

This subroutine solves the forward problem of a sulfate poor, sodium rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NH_4Cl and Na_2SO_4 . The following species are possible:

- Solids: Na_2SO_4
- Aqueous phase: Present.

The iterative solution algorithm of SUBROUTINE CALCH2 is used, with the difference that NaNO_3 , NaCl and NH_4Cl do not precipitate.

SUBROUTINE CALCH6

CHAPTER 5: SUBROUTINE REFERENCE

This subroutine solves the forward problem of a sulfate poor, sodium rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is above the DRH of Na_2SO_4 .

The following species are possible:

- Solids: None
- Aqueous phase: Present.

The iterative solution algorithm of SUBROUTINE `CALCH2` is used, with the difference that no salts precipitate.

SUBROUTINE CALC11

This subroutine solves the forward problem of a sulfate rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is below the DRH of NH_4HSO_4 . The following species are possible:

- Solids: Na_2SO_4 , $(\text{NH}_4)_2\text{SO}_4$, NaHSO_4 , NH_4HSO_4 , $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$
- Aqueous phase: Present when $\text{RH} \geq \text{MDRH}$, not present when $\text{RH} < \text{MDRH}$

The case of a dry aerosol ($\text{RH} < \text{MDRH}$) is handled by SUBROUTINE `CALC11A`, while the mutual deliquescence region is handled by SUBROUTINE `CALCMDRH`.

SUBROUTINE CALC11A

This subroutine solves the forward problem of a sulfate rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is below the DRH of NH_4HSO_4 . The following species are possible:

- Solids: Na_2SO_4 , $(\text{NH}_4)_2\text{SO}_4$, NaHSO_4 , NH_4HSO_4 , $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$

CHAPTER 5: SUBROUTINE REFERENCE

- Aqueous phase: None

All of the sodium reacts with sulfate to produce Na_2SO_4 and NaHSO_4 . The remaining sulfate reacts with ammonia to form $(\text{NH}_4)_2\text{SO}_4$, $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$ and/or NH_4HSO_4 . HCl and HNO_3 remain in the gas phase.

SUBROUTINE CALC12

This subroutine solves the forward problem of a sulfate rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NH_4HSO_4 and NaHSO_4 . The following species are possible:

- Solids: Na_2SO_4 , $(\text{NH}_4)_2\text{SO}_4$, NaHSO_4 , $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$
- Aqueous phase: Present when NH_4HSO_4 exists (which deliquesces) or when $\text{RH} \geq \text{MDRH}$.

Water not present when $\text{RH} < \text{MDRH}$

The case of a dry aerosol ($\text{RH} < \text{MDRH}$) is solved by SUBROUTINE CALC11A. The mutual deliquescence region is handled by SUBROUTINE CALCMDRH. If NH_4HSO_4 exists, then it is solved by SUBROUTINE CALC12A.

SUBROUTINE CALC12A

This subroutine solves the forward problem of a sulfate rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NH_4HSO_4 and NaHSO_4 . An aqueous phase is presumed to exist. The following species are possible:

- Solids: Na_2SO_4 , $(\text{NH}_4)_2\text{SO}_4$, NaHSO_4 , $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$

CHAPTER 5: SUBROUTINE REFERENCE

- Aqueous phase: Present

The system is solved iteratively, using the amount of dissolved $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$ as the estimation variable. For each estimation, FUNCTION FUNC12A calculates the ionic equilibria in the aqueous phase. The solubility product of $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$ is used as the objective function.

SUBROUTINE CALC13

This subroutine solves the forward problem of a sulfate rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NaHSO_4 and $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$. The following species are possible:

- Solids: Na_2SO_4 , $(\text{NH}_4)_2\text{SO}_4$, $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$
- Aqueous phase: Present when NH_4HSO_4 or NaHSO_4 exists (dissolved) or when $\text{RH} \geq \text{MDRH}$. Water not present when $\text{RH} < \text{MDRH}$

The case of a dry aerosol ($\text{RH} < \text{MDRH}$) is solved by SUBROUTINE CALCH1A. The composition in the mutual deliquescence region is solved by SUBROUTINE CALCMDRH. If NH_4HSO_4 or NaHSO_4 exists, then the case is solved by SUBROUTINE CALC13A.

SUBROUTINE CALC13A

This subroutine solves the forward problem of a sulfate rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NaHSO_4 and $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$. Bisulfate salts are dissolved in an aerosol phase which is presumed to exist. The following species are possible:

CHAPTER 5: SUBROUTINE REFERENCE

- Solids: Na_2SO_4 , $(\text{NH}_4)_2\text{SO}_4$, $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$
- Aqueous phase: Present

The iterative solution algorithm of SUBROUTINE `CALCI2A` is used, with the difference that NaHSO_4 does not precipitate.

SUBROUTINE CALCI4

This subroutine solves the forward problem of a sulfate rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$ and $(\text{NH}_4)_2\text{SO}_4$. The following species are possible:

- Solids: Na_2SO_4 , $(\text{NH}_4)_2\text{SO}_4$
- Aqueous phase: Present

The system is solved iteratively, using the amount of dissolved Na_2SO_4 as an estimation variable. For each estimation, FUNCTION `FUNCI4A` calculates the ionic equilibria in the aqueous phase. The solubility product of Na_2SO_4 is used as the objective function.

SUBROUTINE CALCI5

This subroutine solves the forward problem of a sulfate rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of $(\text{NH}_4)_2\text{SO}_4$ and Na_2SO_4 . The following species are possible:

CHAPTER 5: SUBROUTINE REFERENCE

- Solids: Na₂SO₄
- Aqueous phase: Present

The system is solved iteratively, using the same algorithm as in SUBROUTINE CALCI4. The only difference is that (NH₄)₂SO₄ does not precipitate.

SUBROUTINE CALCI6

This subroutine solves the forward problem of a sulfate rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is above the DRH of Na₂SO₄. The following species are possible:

- Solids: None
- Aqueous phase: Present

The concentrations of species are mainly determined by the $HSO_4^-(aq) \longleftrightarrow H^+(aq) + SO_4^{2-}(aq)$ equilibrium.

SUBROUTINE CALCJ1

This subroutine solves the forward problem of a sulfate super-rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is below the DRH of NH₄HSO₄. The following species are possible:

- Solids: NH₄HSO₄, NaHSO₄
- Aqueous phase: Present

The system is solved iteratively, using the amount of dissolved NaHSO₄ as an estimation variable. For each estimation, FUNCTION FUNCJ1 calculates the dissolved NH₄HSO₄ and the

CHAPTER 5: SUBROUTINE REFERENCE

ionic equilibria in the aqueous phase. The solubility product of NaHSO₄ is used as the objective function.

SUBROUTINE CALCJ2

This subroutine solves the forward problem of a sulfate super-rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NH₄HSO₄ and NaHSO₄. The following species are possible:

- Solids: NaHSO₄
- Aqueous phase: Present

The system is solved iteratively, using the amount of dissolved NaHSO₄ as an estimation variable. For each estimation, FUNCTION FUNCJ2 calculates the ionic equilibria in the aqueous phase. The solubility product of NaHSO₄ is used as the objective function.

SUBROUTINE CALCJ3

This subroutine solves the forward problem of a sulfate super-rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is above the DRH of NaHSO₄. The following species are possible:

- Solids: None
- Aqueous phase: Present

Speciation is mainly controlled by the $HSO_{4(aq)}^- \longleftrightarrow H_{(aq)}^+ + SO_{4(aq)}^{2-}$ equilibrium.

SUBROUTINE CALCK1

CHAPTER 5: SUBROUTINE REFERENCE

This subroutine solves the reverse problem of a sulfate poor, ammonium-sulfate aerosol system, when the relative humidity is below the DRH of $(\text{NH}_4)_2\text{SO}_4$. The following species are possible:

- Solids: $(\text{NH}_4)_2\text{SO}_4$
- Aqueous phase: Not present

Solid $(\text{NH}_4)_2\text{SO}_4$ is calculated from the amount of sulfates available. Gas phase ammonia in this case is undefined (it is a free parameter).

SUBROUTINE CALCK2

This subroutine solves the reverse problem of a sulfate poor, ammonium-sulfate aerosol system, when the relative humidity is above the DRH of $(\text{NH}_4)_2\text{SO}_4$. The following species are possible:

- Solids: None
- Aqueous phase: Present

The concentrations in this case are mainly determined by the equilibria: $\text{NH}_{3(g)} \xleftarrow{K_{21}} \text{NH}_{3(aq)}$

and $\text{NH}_{3(aq)} + \text{H}_2\text{O}_{(aq)} \xleftarrow{K_{22}} \text{NH}_{4(aq)}^+ + \text{OH}_{(aq)}^-$.

SUBROUTINE CALCN1

This subroutine solves the reverse problem of a sulfate poor, ammonium-nitrate-sulfate aerosol system, when the relative humidity is below the DRH of NH_4NO_3 . The following species are possible:

CHAPTER 5: SUBROUTINE REFERENCE

- Solids: $(\text{NH}_4)_2\text{SO}_4$, NH_4NO_3
- Aqueous phase: Present when $\text{RH} \geq \text{MDRH}$, not present when $\text{RH} < \text{MDRH}$

The case of a dry aerosol ($\text{RH} < \text{MDRH}$) is solved by SUBROUTINE CALCN1A. The composition in the mutual deliquescence region is solved by SUBROUTINE CALCMDRP.

SUBROUTINE CALCN1A

This subroutine solves the forward problem of a sulfate poor, ammonium-nitrate-sulfate aerosol system, when the relative humidity is below the DRH of NH_4NO_3 . The following species are possible:

- Solids: $(\text{NH}_4)_2\text{SO}_4$, NH_4NO_3
- Aqueous phase: None

$(\text{NH}_4)_2\text{SO}_4$ is calculated from the sulfates, while NH_4NO_3 is calculated from the nitrates. Species in the gas phase are determined from the $\text{NH}_4\text{NO}_{3(s)} \longleftrightarrow \text{NH}_{3(g)} + \text{HNO}_{3(g)}$ equilibrium.

SUBROUTINE CALCN2

This subroutine solves the reverse problem of a sulfate poor, ammonium-nitrate-sulfate aerosol system, when the relative humidity is between the DRH of NH_4NO_3 and $(\text{NH}_4)_2\text{SO}_4$. The following species are possible:

- Solids: $(\text{NH}_4)_2\text{SO}_4$
- Aqueous phase: Present

CHAPTER 5: SUBROUTINE REFERENCE

The system is solved iteratively, using the amount of dissolved $(\text{NH}_4)_2\text{SO}_4$ as the estimation variable. For each estimation, FUNCTION FUNCN2 calculates the gas phase and ionic species.

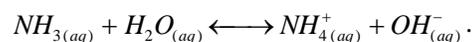
The solubility product of $(\text{NH}_4)_2\text{SO}_4$ is used as the objective function.

SUBROUTINE CALCN3

This subroutine solves the reverse problem of a sulfate poor, ammonium-nitrate-sulfate aerosol system, when the relative humidity is above the DRH of $(\text{NH}_4)_2\text{SO}_4$. The following species are possible:

- Solids: None
- Aqueous phase: Present

The concentrations in this case are determined from the equilibria: $\text{NH}_{3(g)} \longleftrightarrow \text{NH}_{3(aq)}$



SUBROUTINE CALCQ1

This subroutine solves the reverse problem of a sulfate poor, sodium poor, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is below the DRH of NH_4NO_3 .

The following species are possible:

- Solids: $(\text{NH}_4)_2\text{SO}_4$, NH_4NO_3 , NH_4Cl , Na_2SO_4
- Aqueous phase: Present when $\text{RH} \geq \text{MDRH}$, not present when $\text{RH} < \text{MDRH}$

The case of a dry aerosol ($\text{RH} < \text{MDRH}$) is solved by SUBROUTINE CALCQ1A. The composition in the mutual deliquescence region is solved by SUBROUTINE CALCMDRP.

CHAPTER 5: SUBROUTINE REFERENCE

SUBROUTINE CALCQ1A

This subroutine solves the reverse problem of a sulfate poor, sodium poor, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is below the MDRH region of SUBROUTINE CALCQ1. The following species are possible:

- Solids: $(\text{NH}_4)_2\text{SO}_4$, NH_4NO_3 , NH_4Cl , Na_2SO_4
- Aqueous phase: Not present

All of the sodium reacts with sulfates to form Na_2SO_4 . Ammonia then reacts with the remaining sulfate, forming $(\text{NH}_4)_2\text{SO}_4$. The excess NH_3 then reacts with HNO_3 and HCl to form NH_4NO_3 , NH_4Cl , until equilibrium is achieved.

SUBROUTINE CALCQ2

This subroutine solves the reverse problem of a sulfate poor, sodium poor, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NH_4NO_3 and NH_4Cl . The following species are possible:

- Solids: $(\text{NH}_4)_2\text{SO}_4$, NH_4Cl , Na_2SO_4
- Aqueous phase: Present when there is NH_4NO_3 (that deliquesces), or when in a MDRH.

If an aqueous phase is present, SUBROUTINE CALCQ2A is called. When in a MDRH, SUBROUTINE CALCMDRP is called. Otherwise, the aerosol is dry, and SUBROUTINE CALCQ1A is used to calculate the composition.

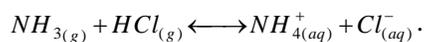
SUBROUTINE CALCQ2A

CHAPTER 5: SUBROUTINE REFERENCE

This subroutine solves the reverse problem of a sulfate poor, sodium poor, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NH_4NO_3 and NH_4Cl . An aqueous phase is presumed to exist. The following species are possible:

- Solids: $(\text{NH}_4)_2\text{SO}_4$, NH_4Cl , Na_2SO_4
- Aqueous phase: Present.

The system is solved iteratively, using gas phase HCl as the estimation variable. For each estimation, FUNCTION FUNCQ2A calculates the ionic equilibria in the aqueous phase, together with the amount of precipitated salts and gas phase NH_3 , HNO_3 . The objective function used is the departure of $\text{Cl}^-_{(\text{aq})}$, $\text{NH}_4^+_{(\text{aq})}$, $\text{HCl}_{(\text{g})}$ and $\text{NH}_3_{(\text{g})}$ from equilibrium, according to the reaction



SUBROUTINE CALCQ3

This subroutine solves the forward problem of a sulfate poor, sodium poor, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NH_4Cl and $(\text{NH}_4)_2\text{SO}_4$. The following species are possible:

- Solids: $(\text{NH}_4)_2\text{SO}_4$, Na_2SO_4
- Aqueous phase: Present when there is some NH_4NO_3 or NH_4Cl (which deliquesces), or when in a MDRH.

If an aqueous phase exists, the problem is solved by SUBROUTINE CALCQ3A. When in a MDRH, SUBROUTINE CALCMDRP is called. Otherwise, the aerosol is dry, and SUBROUTINE CALCQ1A is used to calculate the composition.

CHAPTER 5: SUBROUTINE REFERENCE

SUBROUTINE CALCQ3A

This subroutine solves the reverse problem of a sulfate poor, sodium poor, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NH_4Cl and $(\text{NH}_4)_2\text{SO}_4$. An aqueous phase is presumed to exist. The following species are possible:

- Solids: $(\text{NH}_4)_2\text{SO}_4$, Na_2SO_4
- Aqueous phase: Present.

The system is solved iteratively, using the amount of gas phase HCl as the estimation variable. For each estimation, FUNCTION FUNCQ3A calculates the ionic equilibria in the aqueous phase, together with the amount of precipitated salts and gas phase NH_3 , HNO_3 . The objective function used is the departure of $\text{Cl}^-_{(\text{aq})}$, $\text{NH}_4^+_{(\text{aq})}$, $\text{HCl}_{(\text{g})}$ and $\text{NH}_{3(\text{g})}$ from equilibrium, according to the reaction $\text{NH}_{3(\text{g})} + \text{HCl}_{(\text{g})} \longleftrightarrow \text{NH}_4^+_{(\text{aq})} + \text{Cl}^-_{(\text{aq})}$.

SUBROUTINE CALCQ4

This subroutine solves the reverse problem of a sulfate poor, sodium poor, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of $(\text{NH}_4)_2\text{SO}_4$ and Na_2SO_4 . The following species are possible:

- Solids: Na_2SO_4
- Aqueous phase: Present

The system is solved iteratively, using the amount of gas phase HCl as the estimation variable. For each estimation, FUNCTION FUNCQ4 calculates the ionic equilibria in the aqueous phase, together with the amount of precipitated Na_2SO_4 and gas phase NH_3 , HNO_3 . The objective

CHAPTER 5: SUBROUTINE REFERENCE

function used is the departure of $\text{Cl}^-_{(\text{aq})}$, $\text{NH}_4^+_{(\text{aq})}$, $\text{HCl}_{(\text{g})}$ and $\text{NH}_3_{(\text{g})}$ from equilibrium, according to the reaction $\text{NH}_3_{(\text{g})} + \text{HCl}_{(\text{g})} \longleftrightarrow \text{NH}_4^+_{(\text{aq})} + \text{Cl}^-_{(\text{aq})}$.

SUBROUTINE CALCQ5

This subroutine solves the reverse problem of a sulfate poor, sodium poor, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is above the DRH of Na_2SO_4 .

The following species are possible:

- Solids: None
- Aqueous phase: Present

The solution algorithm of SUBROUTINE CALCQ4 is used, with the difference that Na_2SO_4 does not precipitate.

SUBROUTINE CALCR1

This subroutine solves the reverse problem of a sulfate poor, sodium rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is below the DRH of NH_4NO_3 .

The following species are possible:

- Solids: Na_2SO_4 , NaNO_3 , NaCl , NH_4NO_3 , NH_4Cl
- Aqueous phase: Present when $\text{RH} \geq \text{MDRH}$, not present when $\text{RH} < \text{MDRH}$

The case of a dry aerosol ($\text{RH} < \text{MDRH}$) is solved by SUBROUTINE CALCR1A. The composition in the mutual deliquescence region is calculated by SUBROUTINE CALCMDRP.

SUBROUTINE CALCR1A

CHAPTER 5: SUBROUTINE REFERENCE

This subroutine solves the reverse problem of a sulfate rich, sodium poor, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is below the MDRH region of SUBROUTINE CALCR1. The following species are possible:

- Solids: Na_2SO_4 , NaNO_3 , NaCl , NH_4NO_3 , NH_4Cl
- Aqueous phase: Not present

All of the sodium reacts with the sulfates to produce Na_2SO_4 . The excess sodium reacts to form first NaNO_3 , then NaCl . The remaining HCl and HNO_3 then react with NH_3 to form NH_4NO_3 and NH_4Cl respectively.

SUBROUTINE CALCR2

This subroutine solves the reverse problem of a sulfate poor, sodium rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NH_4NO_3 and NaNO_3 . The following species are possible:

- Solids: Na_2SO_4 , NaNO_3 , NaCl , NH_4Cl
- Aqueous phase: Present when NH_4NO_3 exists or when $\text{RH} \geq \text{MDRH}$, not present when $\text{RH} < \text{MDRH}$

The case of a dry aerosol ($\text{RH} < \text{MDRH}$) is solved by SUBROUTINE CALCR1A. The composition in the mutual deliquescence region is solved by SUBROUTINE CALCMDRP.

SUBROUTINE CALCR2A

This subroutine solves the reverse problem of a sulfate poor, sodium rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NH_4NO_3 and NaNO_3 . An aqueous phase is presumed to exist. The following species are possible:

- Solids: Na_2SO_4 , NaNO_3 , NaCl , NH_4Cl
- Aqueous phase: Present.

The system is solved iteratively, using the amount of gas phase HCl as the estimation variable. For each estimation, FUNCTION `FUNCR2A` calculates the ionic equilibria in the aqueous phase, together with the amount of precipitated salts and gas phase NH_3 , HNO_3 . The objective function used is the departure of $\text{Cl}^-_{(\text{aq})}$, $\text{NH}_4^+_{(\text{aq})}$, $\text{HCl}_{(\text{g})}$ and $\text{NH}_{3(\text{g})}$ from equilibrium, according to the reaction $\text{NH}_{3(\text{g})} + \text{HCl}_{(\text{g})} \longleftrightarrow \text{NH}_4^+_{(\text{aq})} + \text{Cl}^-_{(\text{aq})}$.

SUBROUTINE CALCR3

This subroutine solves the reverse problem of a sulfate poor, sodium rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NaNO_3 and NaCl . An aqueous phase is presumed to exist. The following species are possible:

- Solids: Na_2SO_4 , NaCl , NH_4Cl
- Aqueous phase: Present.

The solution algorithm of SUBROUTINE `CALCR2` is used, with the difference that NaNO_3 does not precipitate.

CHAPTER 5: SUBROUTINE REFERENCE

SUBROUTINE CALCR4

This subroutine solves the reverse problem of a sulfate poor, sodium rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NaCl and NH_4Cl . An aqueous phase is presumed to exist. The following species are possible:

- Solids: Na_2SO_4 , NH_4Cl
- Aqueous phase: Present.

The solution algorithm of SUBROUTINE CALCR2 is used, with the difference that NaNO_3 and NaCl do not precipitate.

SUBROUTINE CALCR5

This subroutine solves the forward problem of a sulfate poor, sodium rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is between the DRH of NH_4Cl and Na_2SO_4 . The following species are possible:

- Solids: Na_2SO_4
- Aqueous phase: Present.

The solution algorithm of SUBROUTINE CALCR2 is used, with the difference that NaNO_3 , NaCl and NH_4Cl do not precipitate.

SUBROUTINE CALR6

This subroutine solves the forward problem of a sulfate poor, sodium rich, sodium-ammonium-nitrate-sulfate-chloride aerosol system, when the relative humidity is above the DRH of Na_2SO_4 .

The following species are possible:

CHAPTER 5: SUBROUTINE REFERENCE

- Solids: None
- Aqueous phase: Present.

The solution algorithm of SUBROUTINE `CALCR2` is used, with the difference that salts do not precipitate.

SUBROUTINE CALCMDRH, CALCMDRP

These subroutines are called when the solution is in a mutual deliquescence relative humidity region. The solution is assumed to be the weighted mean of two states; the “dry” solution (SUBROUTINE `DRYCASE`) and the “saturated liquid” solution (SUBROUTINE `LIQCASE`). SUBROUTINE `CALCMDRH` is used for forward, and `CALCMDRP` for reverse problems.

SUBROUTINE CALCMR

This subroutine calculates the aerosol water content from the ZSR correlation.

SUBROUTINE PUSHERR

This subroutine saves an error message in the error stack.

SUBROUTINE ERRSTAT

This subroutine reports error messages to unit “IO”.

SUBROUTINE KMFUL

Calculates binary activity coefficients by Kusik-Meissner method. Performs real-time calculations.

CHAPTER 5: SUBROUTINE REFERENCE

SUBROUTINE KMTAB

Obtains Kusik-Meissner binary activity coefficients from precalculated tables. The computations have been performed for a wide range of ionic activities and temperatures. SUBROUTINE KMTAB does not access the activity coefficient tables itself but rather calls the appropriate SUBROUTINE KMXXX (XXX being a temperature) to obtain the data.

SUBROUTINE KM198, KM223, KM248, KM273, KM298, KM323

Obtains Kusik-Meissner binary activity coefficients from precalculated tables, for a given temperature (e.g., KM198 is for 198K and KM223 is for 223K). The computations have been performed for a wide range of ionic activities, and the results are stored in lookup tables. These tables are initialized by BLOCK DATA KMCF198, KMCF223, KMCF248, KMCF273, KMCF298, KMCF323.

6. Description of Global Variables

This paragraph presents the most important global variables of ISORROPIA. These variables are allocated in the COMMON blocks defined in include file ISRPIA.INC. They are presented according to the sequence in which the blocks are allotted.

PARAMETERS

NCOMP	Number of input concentrations (=5)
NIONS	Number of ionic species in aqueous phase (=7)
NGASAQ	Number of undissociated gases in aqueous phase (=3)
NSLDS	Number of solids in solid phase (=9)
NPAIR	Number of salts dissolved in aqueous phase (=13)
NERRMAX	Size of error stack (=25)

Input Data

W	Total concentrations (gas+aerosol) of precursors, moles m ⁻³ air W(1) : sodium W(2) : sulfate W(3) : ammonium W(4) : nitrate W(5) : chloride
WAER	Concentrations of precursors in aerosol phase, moles m ⁻³ air. The structure is identical to array W.
TEMP	Ambient temperature, K
RH	Ambient relative humidity (0-1 scale)
IPROB	Type of problem solved 0: Forward ; total concentrations in array W are known 1: Reverse ; aerosol concentrations in array WAER are known
METSTBL	Physical state of aerosol 0: Soluble salts precipitate when supersaturation occurs. 1: Soluble salts do not precipitate when supersaturation occurs- aerosol is in metastable state

CHAPTER 6: GLOBAL VARIABLE REFERENCE

Deliquescence Relative Humidities

DRH2SO4	Pure H ₂ SO ₄
DRNH42S4	Pure (NH ₄) ₂ SO ₄
DRNAHSO4	Pure NaHSO ₄
DRNACL	Pure NaCl
DRNANO3	Pure NaNO ₃
DRNA2SO4	Pure Na ₂ SO ₄
DRNH4HS4	Pure NH ₄ HSO ₄
DRLC	Pure(NH ₄) ₃ H(SO ₄) ₂
DRNH4NO3	Pure NH ₄ NO ₃
DRNH4CL	Pure NH ₄ Cl
DRMLCAB	(NH ₄) ₃ H(SO ₄) ₂ - NH ₄ HSO ₄ mixture
DRMLCAS	(NH ₄) ₃ H(SO ₄) ₂ -(NH ₄) ₂ SO ₄ mixture
DRMASAN	(NH ₄) ₂ SO ₄ - NH ₄ NO ₃ mixture
DRMG1	(NH ₄) ₂ SO ₄ - NH ₄ NO ₃ - Na ₂ SO ₄ - NH ₄ Cl mixture
DRMG2	(NH ₄) ₂ SO ₄ - Na ₂ SO ₄ - NH ₄ Cl mixture
DRMG3	(NH ₄) ₂ SO ₄ - Na ₂ SO ₄ mixture
DRMH1	Na ₂ SO ₄ - NaNO ₃ - NaCl - NH ₄ NO ₃ - NH ₄ Cl mixture
DRMH2	Na ₂ SO ₄ - NaNO ₃ - NaCl - NH ₄ Cl mixture
DRMI1	(NH ₄) ₃ H(SO ₄) ₂ - NH ₄ HSO ₄ - NaHSO ₄ - Na ₂ SO ₄ - (NH ₄) ₂ SO ₄ mixture
DRMI2	(NH ₄) ₃ H(SO ₄) ₂ -NaHSO ₄ - Na ₂ SO ₄ - (NH ₄) ₂ SO ₄ mixture
DRMI3	(NH ₄) ₃ H(SO ₄) ₂ - Na ₂ SO ₄ - (NH ₄) ₂ SO ₄ mixture
DRMQ1	Na ₂ SO ₄ - (NH ₄) ₂ SO ₄ - NH ₄ NO ₃ mixture
DRMR1	Na ₂ SO ₄ -NaNO ₃ - NaCl mixture
DRMR2	Na ₂ SO ₄ - NaCl mixture
DRMR3	Na ₂ SO ₄ - NaNO ₃ mixture
DRMR4	Na ₂ SO ₄ - NaCl - NH ₄ Cl mixture
DRMR5	Na ₂ SO ₄ - NH ₄ Cl mixture
DRMR6	Na ₂ SO ₄ - NaNO ₃ - NH ₄ Cl mixture
DRMR7	Na ₂ SO ₄ - NaNO ₃ - NaCl - NH ₄ Cl mixture
DRMR8	Na ₂ SO ₄ - NaCl - NH ₄ Cl mixture

CHAPTER 6: GLOBAL VARIABLE REFERENCE

DRMR9	Na ₂ SO ₄ - NH ₄ NO ₃ mixture
DRMR10	Na ₂ SO ₄ - NaNO ₃ - NH ₄ NO ₃ mixture
DRMR11	Na ₂ SO ₄ - NH ₄ NO ₃ - NaCl - NH ₄ Cl mixture
DRMR12	Na ₂ SO ₄ - NH ₄ NO ₃ - NH ₄ Cl mixture
DRMR13	Na ₂ SO ₄ - NH ₄ NO ₃ - NaNO ₃ - NH ₄ Cl mixture
WFTYP	Weighting factor type for Mutual Deliquescence Regions. 0: Dry aerosol (no MDRH); 1: weighting factor always 0.5; 2: relative-humidity averaged (default)

Liquid Aerosol Phase

MOLAL	Concentration of ionic species in aqueous phase, moles m ⁻³ air. MOLAL (1): H ⁺ MOLAL (2): Na ⁺ MOLAL (3): NH ₄ ⁺ MOLAL (4): Cl ⁻ MOLAL (5): SO ₄ ²⁻ MOLAL (6): HSO ₄ ⁻ MOLAL (7): NO ₃ ⁻
GASAQ	Concentration of undissociated gases in aqueous phase, moles m ⁻³ air. GASAQ (1): NH ₃ GASAQ (2): HCl GASAQ (3): HNO ₃
WATER	Aerosol water content, kg m ⁻³ air
DRYF	Contains information about the physical state of the system. . TRUE . : There is no aqueous phase present . FALSE . : There is an aqueous phase present
GAMA	Activity coefficients for salts dissolved in the aqueous phase. GAMA (01): NaCl GAMA (02): Na ₂ SO ₄ GAMA (03): NaNO ₃ GAMA (04): (NH ₄) ₂ SO ₄ GAMA (05): NH ₄ NO ₃ GAMA (06): NH ₄ Cl GAMA (07): 2H-SO ₄ GAMA (08): H-HSO ₄

CHAPTER 6: GLOBAL VARIABLE REFERENCE

	GAMA (09):	NH_4HSO_4
	GAMA (10):	HNO_3
	GAMA (11):	HCl
	GAMA (12):	NaHSO_4
	GAMA (13):	$(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$
IONIC		Ionic activity of aqueous phase.
EPSACT		Convergence criterion in activity coefficient calculations.
NADJ		Forces the solution obtained to satisfy total mass balance to machine precision. 0: No adjustment is performed 1: Adjust solution
IACALC		Selects how binary activity coefficients are calculated. 0: Full calculations are performed 1: Pre-calculated tables are used
FRST , CALAIN , CALAOU		Logical flags that control the activity coefficient calculations.

Solid Aerosol Phase

CNH42S4	Concentration of solid $(\text{NH}_4)_2\text{SO}_4$, moles m^{-3} air.
CNH4HS4	Concentration of solid NH_4HSO_4 , moles m^{-3} air.
CNACL	Concentration of solid NaCl , moles m^{-3} air.
CNA2SO4	Concentration of solid Na_2SO_4 , moles m^{-3} air.
CNANO3	Concentration of solid NaNO_3 , moles m^{-3} air.
CNH4NO3	Concentration of solid NH_4NO_3 , moles m^{-3} air.
CNH4CL	Concentration of solid NH_4Cl , moles m^{-3} air.
CNAHSO4	Concentration of solid NaHSO_4 , moles m^{-3} air.
CLC	Concentration of solid $(\text{NH}_4)_3\text{H}(\text{SO}_4)_2$, moles m^{-3} air.

Gas Phase

GNH3	Concentration of gas phase NH_3 , moles m^{-3} air.
GHCL	Concentration of gas phase HCl , moles m^{-3} air.
GHNO3	Concentration of gas phase HNO_3 , moles m^{-3} air.

CHAPTER 6: GLOBAL VARIABLE REFERENCE

Equilibrium Constants

XK1	$HSO_4^-(aq) \xrightleftharpoons{K_1} H^+(aq) + SO_4^{2-}(aq)$
XK2	$NH_3(g) \xrightleftharpoons{K_2} NH_4^+(aq) + OH^-(aq)$
XK21	$NH_3(g) \xrightleftharpoons{K_{21}} NH_3(aq)$
XK22	$NH_3(aq) + H_2O(aq) \xrightleftharpoons{K_{22}} NH_4^+(aq) + OH^-(aq)$
XK3	$HCl(g) \xrightleftharpoons{K_3} H^+(aq) + Cl^-(aq)$
XK31	$HCl(g) \xrightleftharpoons{K_{31}} HCl(aq)$
XK32	$HCl(aq) \xrightleftharpoons{K_{32}} H^+(aq) + Cl^-(aq)$
XK4	$HNO_3(g) \xrightleftharpoons{K_4} H^+(aq) + NO_3^-(aq)$
XK41	$HNO_3(g) \xrightleftharpoons{K_{41}} HNO_3(aq)$
XK42	$HNO_3(aq) \xrightleftharpoons{K_{42}} H^+(aq) + NO_3^-(aq)$
XK5	$Na_2SO_4(s) \xrightleftharpoons{K_5} 2Na^+(aq) + SO_4^{2-}(aq)$
XK6	$NH_4Cl(s) \xrightleftharpoons{K_6} NH_3(g) + HCl(g)$
XK7	$(NH_4)_2SO_4(s) \xrightleftharpoons{K_7} 2NH_4^+(aq) + SO_4^{2-}(aq)$
XK8	$NaCl(s) \xrightleftharpoons{K_8} Na^+(aq) + Cl^-(aq)$
XK9	$NaNO_3(s) \xrightleftharpoons{K_9} Na^+(aq) + NO_3^-(aq)$
XK10	$NH_4NO_3(s) \xrightleftharpoons{K_{10}} NH_3(g) + HNO_3(g)$
XK11	$NaHSO_4(s) \xrightleftharpoons{K_{11}} Na^+(aq) + HSO_4^-(aq)$
XK12	$NH_4HSO_4(s) \xrightleftharpoons{K_{12}} NH_4^+(aq) + HSO_4^-(aq)$
XK13	$(NH_4)_3H(SO_4)_2(s) \xrightleftharpoons{K_{13}} 3NH_4^+(aq) + HSO_4^-(aq) + SO_4^{2-}(aq)$
XK14	$NH_4Cl(s) \xrightleftharpoons{K_{14}} NH_4^+(aq) + Cl^-(aq)$
XKW	$H_2O(aq) \xrightleftharpoons{K_w} H^+(aq) + OH^-(aq)$

Molecular Weights

WMW	Molecular weights of substances in array W or WAER.
	WMW (1) : 23 (sodium)
	WMW (2) : 98 (sulfate)
	WMW (3) : 17 (ammonia)
	WMW (4) : 63 (nitrate)
	WMW (5) : 36.5 (chloride)

CHAPTER 6: GLOBAL VARIABLE REFERENCE

SMW Molecular weights of substances in array MOLALR.

SMW(01) : NaCl
SMW(02) : Na₂SO₄
SMW(03) : NaNO₃
SMW(04) : (NH₄)₂SO₄
SMW(05) : NH₄NO₃
SMW(06) : NH₄Cl
SMW(07) : 2H-SO₄
SMW(08) : H-HSO₄
SMW(09) : NH₄HSO₄
SMW(10) : HNO₃
SMW(11) : HCl
SMW(12) : NaHSO₄
SMW(13) : (NH₄)₃H(SO₄)₂

IMW Molecular weights of ions in array MOLAL.

IMW(01) : H⁺
IMW(02) : Na⁺
IMW(03) : NH₄⁺
IMW(04) : Cl⁻
IMW(05) : SO₄⁻²
IMW(06) : HSO₄⁻
IMW(07) : NO₃⁻

General Variables

VERSION Version information of ISORROPIA

GREAT A very large number

TINY ,
TINY2 Very small numbers

7. Examples of using ISORROPIA

7.1. Interactive demo

This example demonstrates the execution of ISORROPIA when compiled and linked with the interactive driver. For an interactive run to take place, an empty line (“carriage return”) is given when the data input file name is requested. The configuration file used is shown in List 4. The run session, as it would take place, is shown in List 5. The driver first requests the input file name. When a carriage return is given, an interactive session is assumed. Concentration units, type of problem solved, precursor concentrations and ambient conditions are then requested. Results obtained from the session of List 5 are shown in List 6.

List 4: The configuration file ISORROPIA.CNF used.

```
1. *** [ INPUT/OUTPUT PARAMETERS ] ***
2. CREATE REPORT IN FILE? (.T.=yes, .F.=no, on screen)
3. .T.
4. CREATE SPREADSHEET FILE? (.T.=yes, .F.=no)
5. .T.
6.
7. *** [ SOLUTION CONTROL ] ***
8. Convergence criterion (DEFAULT:1.D-6)
9. 1.D-10
10. Max iterations for bisection method (DEFAULT:100)
11. 100
12. Sweeps for activity coef. calculation (DEFAULT:4)
13. 10
14. Accuracy in activity coef. calculation (DEFAULT:5e-2)
15. 1e-2
16. Subdivisions for root tracking (DEFAULT:5)
17. 5
18. Method for binary activity coefs (0-online, 1-tables) (DEFAULT:1)
19. 0
20. Force mass conservation to machine precision? (0-no, 1=yes) (DEFAULT:0)
21. 0
```

List 5: A sample interactive session with ISORROPIA.

```
1. -----
2. ISORROPIA v1.7 (03/26/07)
3. Copyright 1996-2006
4. University of Miami, Carnegie Mellon University,
5. Georgia Institute of Technology
6. Written by Athanasios Nenes (nenes@eas.gatech.edu)
7. Updated by Christos Fountoukis
8. (Christos.Fountoukis@chbe.gatech.edu)
9. -----
10.
```

CHAPTER 7: EXAMPLES OF USING ISORROPIA

```
11. Parameters read from file [isrpio.cnf]
12. File name with runs [Enter=screen input]:
13.
14. Input units (0=umol/m3 air, 1=ug/m3 air) [1.]: 0
15. Problem type? (0=forward, 1=reverse) [0.]: 0
16. Aerosol state? (0=solid+liquid possible, 1=metastable) [0.]: 0
17. Natot (umol/m3 air)? [1.000]: 2.1
18. SO4tot (umol/m3 air)? [1.000]: 1.0
19. NH3tot (umol/m3 air)? [1.000]: 1.0
20. NO3tot (umol/m3 air)? [1.000]: 1.0
21. Cltot (umol/m3 air)? [1.000]: 1.0
22. RH (0-1)? [0.900]: 0.9
23. Temperature (K)? [ 298.0]: 298.0
24.
25. Results saved in file [SCREEN.txt]
26. Data saved in file [SCREEN.dat]
```

List 6: Results of ISORROPIA run.

```
*** [ INPUT ] ***

TOTAL:          (ug/m3 air)    (umol/m3 air)
[Na ]           4.830E+01      2.100E+00
[H2SO4]         9.800E+01      1.000E+00
[NH3 ]          1.700E+01      1.000E+00
[HNO3 ]         6.300E+01      1.000E+00
[HCL ]          3.650E+01      1.000E+00

TEMPERATURE (K) [298.00]
REL. HUMIDITY   [ 90.00] %

*** [ SOLUTION ] ***

GENERAL INFO:
PROBLEM TYPE    [FOREWARD]
AEROSOL STATE   [SOLID + LIQUID POSSIBLE ]
ACTIVITY COEFS [KUSSIK-MEISSNER; FULL CALCULATIONS]
MASS CONSERV.  [NORMAL SOLUTION PROCEDURE]
REGIME CASE     [H5]
SULFATE RATIO  [ 3.100E+00]
SODIUM RATIO   [ 2.100E+00]
SUL.POOR RATIO [ 2.000E+00]

COMPOSITION:
GAS+AEROSOL:   (ug/m3 air)    (umol/m3 air)
[Na ]           4.830E+01      2.100E+00
[H2SO4]         9.800E+01      1.000E+00
[NH3 ]          1.700E+01      1.000E+00
[HNO3 ]         6.300E+01      1.000E+00
[HCL ]          3.650E+01      1.000E+00

GAS:           (ug/m3 air)    (umol/m3 air)    (% mass)    (% mole)
[NH3 ]         7.791E-02      4.583E-03        0.197      0.535
[HNO3 ]        1.988E+01      3.156E-01        50.306     36.862
[HCL ]         1.956E+01      5.360E-01        49.497     62.602

AEROSOL MASS:  (ug/m3 air)    (% of total)
[TOTAL ]       8.637E+02
[LIQUID]       8.637E+02      100.000

NO SOLID AEROSOL PHASE

LIQUID AEROSOL: (ug/m3 air)    (umol/m3 air)    (% mass)    (% mole)
[WATER ]       6.426E+02      3.570E+01        74.393     87.157
[H+ ]          1.653E-02      1.653E-02         0.002      0.040
```

CHAPTER 7: EXAMPLES OF USING ISORROPIA

```
[Na+ ]          4.830E+01    2.100E+00    5.592    5.127
[NH4+ ]         1.792E+01    9.954E-01    2.074    2.430
[Cl- ]          1.647E+01    4.640E-01    1.907    1.133
[NO3- ]         4.243E+01    6.844E-01    4.913    1.671
[SO4-- ]        9.250E+01    9.635E-01   10.709    2.352
[HSO4- ]        3.540E+00    3.649E-02    0.410    0.089
[NH3aq ]         0.000E+00    0.000E+00    0.000    0.000
[HClaq ]         0.000E+00    0.000E+00    0.000    0.000
[HNO3aq]         0.000E+00    0.000E+00    0.000    0.000
[pH ]           1.590E+00
[ IONIC STRENGTH] 6.490E+00
```

```
*** [ MASS BALANCE (% ERROR) ] ***
```

```
Na  TOTAL [ 0.000E+00]
SO4 TOTAL [ 2.118E-14]
NH4 TOTAL [ 0.000E+00]
NO3 TOTAL [ 0.000E+00]
CL  TOTAL [ 0.000E+00]
```

```
*** [ CHARGE BALANCE (umole/m3) ] ***
```

```
TOTAL POSITIVE [ 3.112E+00]
TOTAL NEGATIVE [ 3.112E+00]
```

```
*** [ ERROR MESSAGES ] ***
```

```
NO ERRORS DETECTED
```

```
=====
```

7.2. Batch mode demo

List 7 displays a sample input file, for running ISORROPIA in batch mode. The user specifies the name of the input file during runtime.

List 7: Sample input file for running ISORROPIA in batch mode.

```
1. Input units (0=umol/m3, 1=ug/m3)
2. 0
3.
4. Problem type (0=forward, 1=reverse); Phase state (0=solid+liquid, 1=metastable)
5. 0, 0
6.
7. NH4-SO4 system case
8. Na SO4 NH3 NO3 Cl RH TEMP
9. 0 0.3787878 0.75757575 0.0 0.0 0.5 280.0
```

7.3. Generating customized output

This section displays an example of generating customized output via SUBROUTINE USROUT. List 9 shows how SUBROUTINE USROUT would look like, if the amount of dissolved undissociated ammonia (% w/w) and the partial pressure of gas phase ammonia (in mm Hg) are calculated. The first section (ICODE=1) opens the output file ('diss.dat') and writes a header. The second section (ICODE=2) calculates dissolved and gas phase ammonia (in appropriate units) and writes it to 'diss.dat'. The third section (ICODE=3), closes 'diss.dat' and displays a message on screen.

List 9: Example of SUBROUTINE USROUT .

```
1. C=====
2. C
3. C *** SUBROUTINE USROUT
4. C *** THIS SUBROUTINE IS USED TO GENERATE CUSTOMIZED OUTPUT
5. C
6. C=====
7. C
8.     SUBROUTINE USROUT (ICODE)
9.     INCLUDE 'isrpio.inc'
10.    INCLUDE 'main.inc'
11. C
12. C INITIAL OUTPUT
13. C
14.     IF (ICODE.EQ.1) THEN
15.     OPEN (99, FILE='diss.dat', STATUS='UNKNOWN', RECL=2048)
16.     WRITE(99,*) '"NH3(%ww)" "PNH3(mmHg)"`
17. C
18. C OUTPUT AFTER EACH RUN
19. C
20.     ELSE IF (ICODE.EQ.2) THEN
21.     GG = 2.0*(MOLAL(6)+MOLAL(5)) + MOLAL(7) + MOLAL(4) + GASAQ(2) +
22.     & GASAQ(3) - MOLAL(2) - MOLAL(3) - GASAQ(1)
23.     FRNH3 = 0.0
24.     IF (GG.LT.TINY) FRNH3 = GASAQ(1)
25.     WRITE (99,100) FRNH3*17.0/WATER/10.0, GNH3*TEMP*R*760.0
26. 100  FORMAT (1x, 2(1x, 1PE10.3))
27. C
28. C FINAL OUTPUT
29. C
30.     ELSE IF (ICODE.EQ.3) THEN
31.     CLOSE (99, STATUS='KEEP')
32.     WRITE(*,*) 'Data    saved in file [diss.dat]'
33.     ENDF
34.     RETURN
35.     END
```

8. EXAMPLES OF PROGRAMMING ISORROPIA

8.1. Using the full interface

List 8 displays an example of using the full programming interface of ISORROPIA. First, SUBROUTINE SETPARAM is used to specify the convergence criterion (10^{-6}) and the maximum iterations for all iterative procedures (10). The remaining parameters are assigned negative values, so remain unchanged. ISORROPIA is then called (input values are assumed to be assigned elsewhere in the code), and subsequently the error system is accessed. If any errors occurred during the run, all of them are reported (to unit IO, which is assumed to be connected to a file via an OPEN statement). The program terminates if a fatal error has occurred.

NOTE: If a fatal error has occurred, it is only one (fatal interrupt calculation) and it is the last error found in the error stack.

List 8: Example of using the full interface of ISORROPIA.

```
1. ( . . . )
2.
3. C
4. C SETUP SOLUTION PARAMETERS
5.     CALL SETPARAM (-1d0, -1, 1d-6, 10, -1, -1d0, -1, -1) ! Only EPSI, NSWEEP are set
6. C
7. C CALL ISORROPIA
8.     CNTRL(1) = 0           ! 0=FORWARD PROBLEM, 1=REVERSE PROBLEM
9.     CNTRL(2) = 1         ! 0=SOLID+LIQUID AEROSOL, 1=METASTABLE
10.    CALL ISORROPIA (WI, RHI, TEMPI, CNTRL,
11.    &                W, GAS, AERLIQ, AERSLD, SCASE, OTHER)
12. C
13. C ACCESS ERROR SYSTEM
14.    CALL ISERRINF (ERRSTKI, ERRMSGI, NOFER, STKOFI) ! Obtain error stack
15.    WRITE (IO,*) '*** [ ERROR MESSAGES ] ***'      ! Write error messages
16.    IF (NOFER.EQ.0) THEN                            ! No errors
17.        CALL ERRSTAT (IO, 0, ' ')
18.    ELSE
19.        DO 60 I=1,NOFER                               ! Write error message for
20.            WRITE (IO,*) ' '                          ! each error found in the
21.            CALL ERRSTAT (IO, ERRSTKI(I), ERRMSGI(I)) ! error stack
22. 60    CONTINUE
23.    ENDIF
24.    IF(ERRSTKI(NOFER).GT.1000) STOP 'Fatal error'    ! Fatal error, ABORT.
25.
26. ( . . . )
```

8.2. Incorporation into an air quality (or global) model

This section shows how ISORROPIA can be incorporated into an air quality (or global) model. In the example provided (List 10), we assume that the host model has a subroutine especially for interfacing with a thermodynamic model; this interface is SUBROUTINE ISRPINTR, and interacts with ISORROPIA in three stages. First, the total concentration of species are obtained from the host model (sodium, sulfate, nitrate, ammonium and chloride) and saved in array WI, after converting to mole m⁻³ units. Second, ISORROPIA is called using WI and the ambient RH, T as input. Finally, results are passed back to the main program, expressed in appropriate units (here, concentrations in the host model are expressed in µg m⁻³ for aerosol and partial pressure for gas species). This routine does not setup ISORROPIA solution parameters (through SUBROUTINE SETPARM), so the default parameters are used. Finally, the error system is not accessed (through SUBROUTINE ISERRINF and SUBROUTINE ERRSTAT). Both of these tasks, if needed, could be done somewhere else in the host model.

List 10: A sample interface between ISORROPIA and an air quality model.

```
1. C=====
2. C
3. C *** SUBROUTINE ISRPINTR
4. C *** THIS SUBROUTINE IS A SAMPLE INTERFACE BETWEEN AN AIRSHED MODEL
5. C *** AND ISORROPIA.
6. C
7. C=====
8. C
9. C     SUBROUTINE ISRPINTR(q, c, ps)
10. C
11. C     INCLUDE 'airshed.inc'           ! airshed model declarations
12. C     DOUBLE PRECISION WI(5), GAS(3), AERLIQ(12), AERSLD(9), CNTRL(2),
13. C     &                               WT(5), OTHER(6), RHI, TEMPI
14. C     CHARACTER SCASE*15
15. C
16. C *** CONVERT INPUT CONCENTRATIONS TO moles/m3 *****
17. C
18. C     WI(1) = q(Na )/emw(Kna )*1.D-6
19. C     WI(2) = q(SO4)/emw(KSO4)*1.D-6
20. C     WI(3) = q(NH4)/emw(KNH4)*1.D-6
21. C     WI(4) = q(NO3)/emw(KNO3)*1.D-6
22. C     WI(5) = q(CL )/emw(KCL )*1.D-6
23. C
24. C *** CALL ISORROPIA *****
25. C
```

CHAPTER 8: EXAMPLES OF PROGRAMMING ISORROPIA

```
26.      RHI      = rh          ! Relative humidity provided by airshed model
27.      TEMPI    = temp       ! Temperature (K) provided by airshed model
28.      CNTRL(1) = 0          ! 0=FORWARD PROBLEM, 1=REVERSE PROBLEM
29.      CNTRL(2) = 0          ! 0=SOLID+LIQUID AEROSOL, 1=METASTABLE
30. C
31.      CALL ISORROPIA (WI, RHI, TEMPI, CNTRL,
32.      &              WT, GAS, AERLIQ, AERSLD, SCASE, OTHER)
33. C
34. C *** CONVERT OUTPUT CONCENTRATIONS TO ug/m3 *****
35. C
36. C AEROSOL SPECIES
37. C
38.      q(H2O)    = AERLIQ(8)*18.D6          ! H2O
39.      C(H2O)    = q(H2O)                   ! H2O
40.      C(HP)     = AERLIQ(01)                *1.D6 ! H+(aq)
41.      C(NAP)    = AERLIQ(02)*intmw(03)*1.D6 ! Na+(aq)
42.      C(NH4P)   = AERLIQ(03)*intmw(04)*1.D6 ! NH4+(aq)
43.      C(CLM)    = AERLIQ(04)*intmw(05)*1.D6 ! Cl-(aq)
44.      C(SO4M)   = AERLIQ(05)*intmw(06)*1.D6 ! SO4--(aq)
45.      C(HSO4M)  = AERLIQ(06)*intmw(07)*1.D6 ! HSO4-(aq)
46.      C(NO3M)   = AERLIQ(07)*intmw(08)*1.D6 ! NO3-(aq)
47.      C(SC)     = AERSLD(03)*intmw(09)*1.D6 ! NaCl(s)
48.      C(SS)     = AERSLD(05)*intmw(10)*1.D6 ! Na2SO4(s)
49.      C(SN)     = AERSLD(01)*intmw(11)*1.D6 ! NaNO3(s)
50.      C(AS)     = AERSLD(06)*intmw(12)*1.D6 ! (NH4)2SO4(s)
51.      C(AN)     = AERSLD(02)*intmw(13)*1.D6 ! NH4NO3(s)
52.      C(AC)     = AERSLD(04)*intmw(14)*1.D6 ! NH4Cl(s)
53.      C(SA)     = 0.0D0                    ! H2SO4(aq)
54.      C(AB)     = AERSLD(08)*intmw(16)*1.D6 ! NH4HSO4(s)
55.      C(SB)     = AERSLD(07)*intmw(17)*1.D6 ! NaHSO4(s)
56.      C(LC)     = AERSLD(09)*intmw(18)*1.D6 ! (NH4)4H(SO4)2(s)
57. C
58. C GASEOUS SPECIES
59. C
60.      ps(k,INH3 ) = GNH3*rgas*temp          ! NH3(g)   (Pa)
61.      ps(k,IHNO3 ) = GHNO3*rgas*temp        ! HNO3(g)  (Pa)
62.      ps(k,IH2SO4) = 0.0d0                  ! H2SO4(g) (Pa)
63.      ps(k,IHCL ) = GHCL*rgas*temp          ! HCL(g)   (Pa)
64. C
65.      RETURN
66. C
67. C *** END OF SUBROUTINE ISRPNTR *****
68. C
69.      END
```

9. References

Nenes A, Pandis SN, Pilinis C (1998a). ISORROPIA: A new thermodynamic equilibrium model for multiphase multicomponent inorganic aerosols, *Aquat.Geoch.*, **4**, 123-152

Nenes A., Pilinis C., and Pandis S.N. (1998b) Continued Development and Testing of a New Thermodynamic Aerosol Module for Urban and Regional Air Quality Models, *Atmos. Env.*, **33**, 1553-1560

Ansari AS, Pandis SN (1999a) An analysis of four models predicting the partitioning of semivolatile inorganic aerosol components, *Aeros.Sci.Tech.*, **31**, 129-153

Ansari AS, Pandis SN (1999b) Prediction of multicomponent inorganic atmospheric aerosol behavior, *Atmos.Env.*, **33**, 745-757

APPENDIX A. Changes between versions of ISORROPIA

Changes between versions 1.5 and 1.7

- A bug in the calculation of vapor-phase nitric acid concentration for reverse ammonium-sulfate-nitrate-water problems (sulfate super-rich aerosol only) was corrected.
- SUBROUTINE ADJUST is added to provide the user with the option to “force” ISORROPIA to conserve mass up to machine precision.
- The variable NADJ is added to the COMMON block to control whether SUBROUTINE ADJUST is to be called after every run or not.
- Inaccuracy in the polynomial fit used to produce the water activity database would sometimes give too high water content for relative humidity near saturation (RH>99.4%). This problem has been corrected.
- SUBROUTINE CALCMR: CASE F and CASE C, MOLALR(9) is used instead of MOLALR(4) for computation of liquid water.
- CASE J and SUBROUTINE FUNCG4A, computation of solid Na₂SO₄ has been corrected..
- SUBROUTINE KMFUL, unnecessary computations of the temperature dependence of binary activity coefficients are now avoided.
- The computational efficiency of some “forward” problem routines is improved.
- SUBROUTINE FUNC12A, GAMA(12) is used instead of GAMA(9) for the “A3” constant calculation.

APPENDIX A: CHANGES BETWEEN VERSIONS OF ISORROPIA

- BLOCK DATA KMCFxxxA, KMCFxxxB, and KMCFxxxC (where xxx are one of 198, 223, 248, 273, 298, 323) are now merged into one BLOCK DATA subprogram, to avoid compatibility issues with some FORTRAN compilers. The tabulated Kusik-Meissner binary activity coefficients have been recomputed by calling KMFUL for the midpoint of each ionic strength interval. The code that computes ipos (position in Kusik-Meissner binary activity coefficient lookup tables, as defined in the above block data sections) has been fixed in subroutines KM198, KM223, KM248, KM273, KM298, and KM323.
- SUBROUTINE CACLACT, upper limit in ionic strength is allowed to be 100 (as opposed to 20 in the older versions). The upper limit is implemented to avoid numerical exception errors.
- In *main.for*, a bug in the CALL SETPARM was fixed ("-1d0" in the argument list was changed to the appropriate "-1").
- The water activity database was updated, using the output from the AIM model (<http://www.hpc1.uea.ac.uk/~e770/aim.html>)
- The DRH of NH₄NO₃ is strongly dependent on temperature. Under low temperature conditions (T<270K), this changes the order (starting from low to high RH) with which salts deliquesce. For these cases the DRH of NH₄NO₃ is assumed to not "cross over" the DRH of the other salts present in the solution, especially since thermodynamic data for supercooled NH₄NO₃ solutions are not known. The same is assumed for NH₄Cl and NaNO₃ which exhibit similar behavior with NH₄NO₃.
- Temperature corrections for the water activities have been turned off.
- In subroutines INIT1 and INIT2 unnecessary computations of m_{oi} for salts not present in the solution are now avoided.

Changes between versions 1.4 and 1.5

- When the aerosol water content was very low, SUBROUTINE CALCD2, CALCD3 and CALCHS4 would occasionally generate numerical exception errors. These problems have been fixed.
- SUBROUTINE CALCH1, CALCH2, CALCH3, CALCH4, CALCH5, CALCH6, CALCG3, CALCG4 and CALCG5 have been modified so that NH_4^+ and H^+ concentrations are more accurately determined.
- SUBROUTINE PUSHEND (interactive driver routine) would crash on some platforms whenever an end-of-file (EOF) was encountered. This problem is now fixed.
- Typographical errors in the manual were corrected.

Changes between versions 1.3 and 1.4

- Temperature dependency of the activity coefficients is included. This has been done for both pre-calculated tables and online calculation of activity coefficients.
- The temperature dependence of the activity coefficients is used to correct the water uptake data for temperatures other than 298.15K.
- The MDRH points for all the systems considered in ISORROPIA are updated and complete. For this purpose, the GFEMN model of Ansari and Pandis (1999b) was used to calculate the necessary data.

APPENDIX A: CHANGES BETWEEN VERSIONS OF ISORROPIA

- The computational efficiency of the activity coefficient routines and of some “forward” problem routines is improved.
- The interactive driver has changed. The input file name is now specified at runtime, and not specified in the configuration file `ISRPIA.CNF` (as done in previous versions). Furthermore, the name of the input file is used to generate the names of the output files.
- ISORROPIA v1.1 through 1.3 was referred to occasionally as ISORROPIA-plus (The “plus” was added initially to distinguish the code from ISORROPIA version 1.0). To avoid confusion, the “plus” suffix is omitted from all versions of ISORROPIA (starting from 1.4).

Changes between versions 1.1 and 1.3

- An error in the code that decreased the amount of aerosol nitrate and chloride (when the sulfate ratio is exactly 2.0) is corrected.
- The water activity data is updated with the most recent datasets available. As a consequence, the water uptake calculated by previous versions of the code is slightly different.
- Errors in the filesystem of the interactive driver are corrected.
- In “reverse problems”, the MDRH weighting algorithm was occasionally not weighing the dry solution correctly. This problem is now corrected.

Changes between versions 1.0 and 1.1

- The aerosol can either be in a thermodynamically stable or a metastable state. In the first case, salts precipitate when supersaturation occurs, while in the second, the aerosol is always an aqueous solution (with no precipitate).
- A new solution algorithm was developed for ammonia-sulfate systems that accelerates the solution process.
- A completely new programming interface was developed. Apart from running the model itself, there are now a set of utility subroutines with which one could configure the model, obtain information and access the error system.
- A new error system was developed. There is now an error stack that stores all the errors that appear during a run. The stack is checked for overflow, and there is a routine that prints error messages to a user-defined unit. This error system could be accessed any time after a run has been completed, at any point of a program.
- Backward compatibility with previous versions of ISORROPIA is ensured. Call interfaces from older versions are supported, thus allowing the modeler to use newer versions of the code, without any additional programming effort.

APPENDIX B. Older interfaces of ISORROPIA

ISORROPIA supports calling interfaces of older versions for the sake of backward compatibility. This allows a modeler to use newer versions of the code, without any additional programming effort. The interfaces of previous versions are described below.

Interface for versions 1.1 to 1.6

These versions use the interface of version 1.7

Interface for version 1.0

This version of the model is accessed through SUBROUTINE ISOPLUS (found in file ISOCOM.FOR). Metastable aerosol calculations (first introduced in version 1.1) cannot be performed using this interface.

Calling Syntax

```
CALL ISOPLUS (WI, RHI, TEMPI, IPROBI, GAS, AERLIQ, AERSLD, DRYI)
```

Input Variables

- **WI**, DOUBLE PRECISION array of length [5]. Precursor concentrations, expressed in moles m^{-3} air. Depending on the type of problem solved, WI contains either *gas+aerosol* or *aerosol only* concentrations. Each array element contains:

WI(1)	1. sodium
WI(2)	2. sulfate
WI(3)	3. ammonium
WI(4)	4. nitrate
WI(5)	5. chloride

- **RHI**, DOUBLE PRECISION variable. Ambient relative humidity expressed on a (0,1) scale.

APPENDIX B: OLDER INTERFACES OF ISORROPIA

- **TEMPI**, DOUBLE PRECISION variable. Ambient temperature expressed in Kelvins.

- **IPROB**, INTEGER variable. The type of problem solved:

IPROB = 0 Forward problem is solved. In this case, array WI contains GAS and AEROSOL concentrations together.

IPROB = 1 Reverse problem is solved. In this case, array WI contains AEROSOL concentrations only.

Output Variables

- **GAS**, DOUBLE PRECISION array of length [03]. Gaseous species concentrations, expressed in moles m⁻³ air. Each array element contains:

GAS (1) NH₃

GAS (2) HNO₃

GAS (3) HCl

- **AERLIQ**, DOUBLE PRECISION array of length [11]. Liquid aerosol species concentrations, expressed in moles m⁻³ air. Each array element contains:

AERLIQ (01) H⁺_(aq)

AERLIQ (02) Na⁺_(aq)

AERLIQ (03) NH₄⁺_(aq)

AERLIQ (04) Cl⁻_(aq)

AERLIQ (05) SO₄⁻²_(aq)

AERLIQ (06) HSO₄⁻_(aq)

AERLIQ (07) NO₃⁻_(aq)

AERLIQ (08) H₂O

AERLIQ (09) NH_{3(aq)} (undissociated)

AERLIQ (10) HCl_(aq) (undissociated)

AERLIQ (11) HNO_{3(aq)} (undissociated)

APPENDIX B: OLDER INTERFACES OF ISORROPIA

- **AERSLD**, DOUBLE PRECISION array of length [09]. Solid aerosol species concentrations, expressed in moles m^{-3} air. Each array element contains:

AERSLD (01)	$\text{NaNO}_{3(s)}$
AERSLD (02)	$\text{NH}_4\text{NO}_{3(s)}$
AERSLD (03)	$\text{NaCl}_{(s)}$
AERSLD (04)	$\text{NH}_4\text{Cl}_{(s)}$
AERSLD (05)	$\text{Na}_2\text{SO}_{4(s)}$
AERSLD (06)	$(\text{NH}_4)_2\text{SO}_{4(s)}$
AERSLD (07)	$\text{NaHSO}_{4(s)}$
AERSLD (08)	$\text{NH}_4\text{HSO}_{4(s)}$
AERSLD (09)	$(\text{NH}_4)_4\text{H}(\text{SO}_4)_2(s)$

- **DRY**, LOGICAL variable. Contains information about the physical state of the system.
 - . TRUE . There is no aqueous phase present
 - . FALSE . There is an aqueous phase present.

APPENDIX C. Compiling ISORROPIA

Compiling with interactive driver

Compiling and linking ISORROPIA (files ISOFWD.FOR, ISOREV.FOR, ISOCOM.FOR) depends on the platform and compiler system used. A typical UNIX f77 compilation command would look like:

```
f77 main.for isocom.for isofwd.for isorev.for -o isorropia
```

The result of this command would be the generation of an executable, “isorropia”. If the UNIX system does not support FORTRAN source files with a .FOR extension (e.g., IBM AIX xlf compilers), the user must then rename all the .FOR files to corresponding .F files (i.e., main.for should be renamed to main.f, etc.).

For Intel-based PC’s under MS-DOS and using the Lahey Fortran 77 compiler (version 5.x), the command sequence should be:

```
f7713 main.for
f7713 isocom.for
f7713 isofwd.for
f7713 isorev.for
386link main.obj isocom.obj isofwd.obj isorev.obj -EXE isoropia.exe
```

the first four commands compile the .FOR files and generate the compiled object files. The fifth command links the object files together and generates the executable ’ISORROPIA.EXE’. Other compilers usually require a similar compile/link procedure.

Compiling with an air quality (or global) model

In this case, the host model has to be compiled and linked only with the ISORROPIA model files (ISOFWD.FOR, ISOREV.FOR, ISOCOM.FOR). A procedure similar to the previous is followed, but for each of the host model and ISORROPIA files.