AN INTERACTIVE CODE (NETPATH) FOR MODELING NET GEOCHEMICAL REACTIONS ALONG A FLOW PATH

By L. Niel Plummer, Eric C. Prestemon, and David L. Parkhurst

---

U.S. GEOLOGICAL SURVEY

Water-Resources Investigations Report 91-4078

U.S. DEPARTMENT OF THE INTERIOR
March 3, 1849

Reston, Virginia

1991

DEPARTMENT OF THE INTERIOR

MANUEL LUJAN, JR., Secretary

U.S. GEOLOGICAL SURVEY

Dallas L. Peck, Director

For additional information write to:

Chief, Branch of Regional Research,
Northeastern Region
U.S. Geological Survey
432 National Center
12201 Sunrise Valley Drive
Reston, Virginia 22092

Copies of this report can be purchased from:

U.S. Geological Survey
Books and Open-File Reports Section
Federal Center, Bldg. 810
P.O. Box 25425
Denver, Colorado 80225

Additional copies of the software described in this report are available from:

National Water Information System
U.S. Geological Survey
437 National Center
Reston, VA 22092

CONTENTS

DISKETTE (Back Pocket)

Diskette containing information file (READ.BAT), source codes, executable files, and data files pertaining to DB and NETPATH, and test problems listed in this report.

AN INTERACTIVE CODE (NETPATH) FOR MODELING <u>NET</u> GEOCHEMICAL REACTIONS ALONG A FLOW <u>PATH</u>

L. Niel Plummer, Eric C. Prestemon, and David L. Parkhurst

ABSTRACT

NETPATH is an interactive Fortran 77 computer program used to interpret net geochemical mass-balance reactions between an initial and final water along a hydrologic flow path. Alternatively, NETPATH computes the mixing proportion of two initial waters and net geochemical reactions that can account for the observed composition of a final water. The program utilizes previously defined chemical and isotopic data for waters from a hydrochemical system. Every possible geochemical mass balance reaction model is examined between selected evolutionary waters for a set of chemical and isotopic constraints, and a set of plausible phases in the system. The calculations are of use in interpreting geochemical reactions, mixing proportions, evaporation and (or) dilution of waters, and mineral mass transfer in the chemical and isotopic evolution of natural and environmental waters. Rayleigh distillation calculations are applied to each mass-balance model that satisfies the constraints to predict carbon, sulfur, and strontium isotopic compositions at the end point, including radiocarbon dating. This report describes the types of problems that can be solved, the methods used to solve problems, and the features available in the program to facilitate these solutions. Examples are presented that demonstrate most of the applications and features of NETPATH. The codes DB and NETPATH can be executed on and IBM PC[1] or compatible microcomputer.

---

[1] The use of trade, brand or product names in this report is for identification purposes only and does not constitute endorsement by the U.S. Geological Survey.

INTRODUCTION

NETPATH is an interactive Fortran 77 computer program used to interpret net geochemical mass-balance reactions between an initial and final water along a hydrologic flow path. The net geochemical mass-balance reaction consists of the masses (per kilogram of water (H_2O)) of plausible minerals and gases that must enter or leave the initial water along the flow path to define the composition of a selected set of chemical and isotopic constraints observed in the final water. If initial waters mix and subsequently react, NETPATH computes the mixing proportion of two initial waters, and net geochemical reactions that can account for the observed composition of a final water. The program uses previously defined chemical and isotopic data for waters from a hydrochemical system. Every possible geochemical mass-balance reaction is examined between selected evolutionary waters for a set of chemical and isotopic constraints, and a set of plausible phases in the system. The calculations are of use in interpreting geochemical reactions, mixing proportions, evaporation and (or) dilution of waters, and mineral mass transfer in the chemical and isotopic evolution of natural and environmental waters. If sufficient isotopic data are available, Rayleigh distillation calculations are applied to each mass-balance model that satisfies the constraints to predict carbon, sulfur, and strontium isotopic compositions at the end-point, including radiocarbon dating. The modeling approach of NETPATH is discussed by Plummer and Back (1980), Parkhurst and others, (1982), Plummer and others, (1983), Plummer (1984), and Plummer and others (1990). The isotopic evolution equations used and a description of radiocarbon dating may be found in Wigley and others, (1978, 1979), Fontes and Garnier (1979), and Wigley and Muller (1981). This report describes the concepts of mass balance reaction modeling included in NETPATH, capabilities of the program, the methods needed for entering and manipulating data, methods for interpreting the results of the program, and limitations of the calculations. Detailed examples are presented that demonstrate most of the features of the code.

NETPATH modeling uses two Fortran 77 codes: DB.FOR, and NETPATH.FOR. DB is a data-base program that allows entering and editing of chemical and isotopic data for a set of water analyses. A modified version of the aqueous speciation program, WATEQF (Plummer and others, 1976) is included in DB to create input files to NETPATH. DB accepts a [*Filename*] which is attached to key files created for the particular set of analyses. The most important of these are the "[*Filename*].LON" file which contains all the hydrochemical data for the set of water analyses and the "[*Filename*].PAT" file which contains all the chemical and isotopic data for the water analyses read by NETPATH. Below is a list of the essential and internal files required to enter hydrochemical data through DB and run NETPATH.

Essential Files

The files listed below are required in compiling and running DB and NETPATH. If it is not necessary to recompile DB and NETPATH, only the files DB.EXE, NETPATH.EXE, NETPATH.DAT, and WATEQF.DAT need to be resident in the directory to run the programs DB and NETPATH. If the file NETPATH.DAT is not found in the directory, NETPATH creates an empty NETPATH.DAT file.

DB.FOR: This is the Fortran 77 source code of the database editing program, DB. The hydrochemical data from field information are entered and saved in the .LON file for later manipulation. DB.FOR also includes the program WATEQFP which is a modified version of ion-association model, WATEQF (Plummer and others, 1976), used to calculate the distribution of aqueous species and construct the input file to NETPATH ([*Filename*].PAT). The DB source code is listed in Attachment A.

DB.EXE: Executable file for DB. The copy of DB.EXE provided with this report was compiled using Microsoft Fortran (rev. 5.0) and is consistent with revisions 4.0 and later. The executable file can be run on microcomputers equipped either with or without a math co-processor.

WATEQF.DAT: This file contains thermodynamic data and information on the aqueous model used to calculate the distribution of species by WATEQFP. For further information see Plummer and others (1976). The WATEQF.DAT file is listed in Attachment B.

NETPATH.FOR: This is the Fortran 77 source code to the main program. This program allows the input data to be used to create mass-balance models between evolutionary water analyses. The NETPATH source code is listed in Attachment C.

NETPATH.BLO: This file contains the declarations of many of the key variables used by NETPATH. It is included when compiling NETPATH.FOR. Comment statements listed in NETPATH.BLO define key variables used in NETPATH. Attachment D is a listing of the file NETPATH.BLO.

NETPATH.EXE: Executable file for NETPATH. The copy of NETPATH.EXE provided with this report was compiled using Microsoft Fortran (rev. 5.0) and is consistent with revisions 4.0 and later. The executable file can be run on microcomputers equipped either with or without a math co-processor.

NETPATH.DAT: This is a file of previously entered phases, their stoichiometries, and default isotopic compositions which is read by NETPATH. These can be used directly or after editing as plausible phases in a reaction model. Attachment E is a listing of the NETPATH.DAT file.

Internal Files

Other files are essential but are created and managed internally by DB and NETPATH. These are listed below ([*Filename*].extension) by extension and name.

Files Created and Managed by DB

.LON: This file contains all of the chemical and isotopic data entered through DB for the particular [*Filename*]. The .LON file is updated through the Save option of DB.

.PAT: This file contains all chemical and isotopic data needed by NETPATH for the particular [*Filename*]. The .PAT file is created by DB after first selecting the Save option and then selecting Quit.

.IN: This is the input file to WATEQFP. The .IN file is deleted by DB after the .PAT file has been written.

.OUT: This is the output file from WATEQFP. The file contains the distribution of species, saturation indices, and other calculations made by WATEQFP (see Plummer and others, 1976 for further details).

DB.FIL: This file contains an updated list of prefix names ([*Filename*]) to all [*Filename*].LON files created by DB.

NETPATH.FIL: This file contains an updated list of prefix names ([*Filename*]) to all [*Filename*].PAT files created by DB. The file is also read by NETPATH. On initiation of NETPATH the list of .PAT files is displayed for selection. All modeling within NETPATH is performed on water analyses from a single .PAT file.

CHECK: This file contains the percent charge imbalance of each water analyses in the .LON file. CHECK is created from the <P>rint option of DB and is an optional selection.

OUTPUTnn: This file contains a report of the water analysis for well number nn. The report is printed from the <P>rint option of DB. Attachment F is an example of this report.

Files Created and Managed by NETPATH

MODEL.FIL: If previous reaction models have been constructed and saved through NETPATH, this file contains a list of the user-defined names of these reaction models associated with the selected .PAT file. Any resident model files are displayed by NETPATH after selection of a new well file (listed in the NETPATH.FIL file). This is accomplished in NETPATH through the <E>dit command.

NETPATH.DAT: This file contains stoichiometries of previously defined phases. Although a preliminary version of NETPATH.DAT is provided with NETPATH, phases in NETPATH.DAT can be added, edited and deleted through NETPATH. The file NETPATH.DAT should reside in the directory containing the NETPATH run module. But if not present, an empty NETPATH.DAT file is created by NETPATH.

NETPATH.OUT: If Rayleigh isotope simulations are invoked through NETPATH, selected data specific to these isotope fractionation calculations are written to NETPATH.OUT.

Model files: Model files can be saved using the Save option of NETPATH. The user-assigned file name is stored in the file MODEL.FIL for subsequent retrieval and editing and (or) running through NETPATH.

Result files: NETPATH results can be saved to user-defined result files using the Save option.

Installation and Setup

The back pocket of this report contains a 1.2M diskette of the essential files listed above. The executable files (.EXE files) and essential data files are in the top-level directory of the diskette. The source codes are in the directories <DB> and <NETPATH> on the diskette. The directory <EXAMPLES> contains the .LON, .PAT, NETPATH.FIL, MODEL.FIL, and model files necessary to run the seven examples and test problems of this report. The programs can be executed from the diskette, but operation is inefficient and the number and size of files generated by DB and NETPATH will soon exceed the available space on the diskette. If the microcomputer is configured with a hard drive, it is recommended that at least the .EXE and data files in the top-level directory of the diskette be copied to the hard drive for execution there. To run the examples and test problems, it is necessary to copy the contents of the directory <EXAMPLES> to the same directory containing the .EXE and

.DAT files found in the top-level directory of the diskette. For further information about the diskette (back pocket), insert the diskette into the A drive of the microcomputer, and at the A> prompt, type READ; or similarily, if the diskette is to be read from the B drive.

DB and NETPATH use screen-control commands that are specific to VT100 terminals. For operation on a microcomputer, the CONFIG.SYS file located in the top-level directory should contain the statement DEVICE=ANSI.SYS, or if the ANSI.SYS file is not located in the top-level directory, the CONFIG.SYS file should show the full path name to the ANSI.SYS file, such as, DEVICE=C:\DOS\ANSI.SYS, if the file is located in the DOS directory on the C drive. If DB and NETPATH are transported to mini-computers or mainframes, the results should be viewed on a VT100 terminal, or a terminal configured to emulate a VT100 terminal.

The executable files of DB and NETPATH were compiled using Microsoft Fortran 5.0 and are consistent with Microsoft Fortran revision 4.0 and greater. For installation on other computers, a few changes to the source codes may be required. Specifically, the $INCLUDE statements may need to be revised to be consistent with other compilers, or alternatively, it may be necessary to replace the $INCLUDE statements with the file NETPATH.BLO at every occurrence in the source codes.

The executable files of DB and NETPATH provided on the diskette (back pocket) were compiled for microcomputers configured with a math co-processor, but will emulate a math co-processor if not present.

Data Needed to Run DB and NETPATH

Certain analytical data are required in every modeling situation. These include temperature, pH, and a name for each analysis (well). Some ion concentrations are almost always needed for realistic calculations. Carbon, calcium, magnesium, sodium, chloride, sulfate are typically, but not always, included in mass-balance modeling. Attachment F is a copy of a report printed by DB showing all the data that can be stored by DB.

IMPORTANT CONCEPTS IN NETPATH

In this section, the key concepts and terminology used in constructing net geochemical mass-balance reactions are reviewed. More extensive discussion is given by Plummer and others (1983) and Plummer (1984). In constructing net geochemical mass-balance reactions, it is necessary to select truly evolutionary initial and final waters, such as waters sampled along a flow path in a confined regional ground-water system, or laboratory waters sampled sequentially from a reactor. In ground-water systems, hydrologic data and hydrogeologic intuition are commonly required in selecting truly evolutionary waters. The danger lies in the fact that regardless of whether waters are truly evolutionary, NETPATH has no means of checking this, and will consequently always report reactions if they can be found. In this report, a net geochemical mass-balance reaction is often referred to as a model. A model is defined as the masses (per kilogram H_2O) of a set of plausible minerals and gases that must enter or leave the initial solution in order to exactly define a set of selected elemental and isotopic constraints observed in a final (evolutionary) water.

Constraints

A constraint is typically an chemical element, but may also be an expression of electron conservation (termed redox state) or conservation of a particular isotope of an element. A constraint is included in the model to constrain the masses of selected phases (minerals and gases) that can enter or leave the aqueous solution. The constraints selected for the model will determine the number and types of phases that need to be selected to solve the modeling problem. The total list of constraints available in NETPATH is--

```
                     List of constraints

 1: Carbon          2: Sulfur          3: Calcium          4: Aluminum
 5: Magnesium       6: Sodium          7: Potassium        8: Chloride
 9: Fluoride       10: Silica         11: Bromide         12: Boron
13: Barium         14: Lithium        15: Strontium       16: Iron
17: Manganese      18: Nitrogen       19: Phosphate       20: Redox
21: Carbon-13      22: C-14 (pmc)     23: Sulfur-34       24: Strontium-87
25: Temperature    26: Deuterium      27: Oxygen-18       28: Tritium
```

If an element is selected that is not contained in any of the chosen phases, the mass balance can be satisfied only by introduction of mixing of initial waters. The redox state, RS, (Parkhurst and others, 1980, 1982; Plummer and others, 1983) provides a means of accounting for electron transfer and must be included for any oxidation-reduction reaction. The isotopic constraints should be included only for true isotope mass-balance problems, such as including deuterium or oxygen-18 to determine the mixing of two water masses, or including carbon-13 when there are only mineral sources along the flow path. Using carbon, sulfur, and strontium isotopes as constraints is realistic only when the elements corresponding to the isotopes (carbon, sulfur, or strontium) are also included as constraints. If both sources and sinks for a particular isotope exist along the flow path, the isotope should not be selected as a constraint.

If sufficient chemical and isotopic data are available, NETPATH can calculate the isotopic composition of the final water for a particular model using the generalized isotope evolution model of Wigley and others (1978). An isotope need not be included as a constraint to be treated in isotopic evolution calculations. Comparing the observed and calculated isotopic compositions at final wells is a means of eliminating reaction models from further consideration. The isotopes carbon-13, carbon-14, sulfur-34, and strontium-87 are often involved in reactions with multiple sources and sinks and are more appropriately treated using the isotope evolution calculations of NETPATH.

Although many trace elements could have been included in NETPATH, interpretation of mass-balance results based on trace element concentrations requires more detailed knowledge of the composition of phases than is usually available. Therefore, many trace element constraints have not been included in NETPATH.

Redox State

The redox state is defined for each aqueous solution and phase considered by NETPATH and is computed according to the conventions originally defined in Parkhurst and others (1980, 1982), and Plummer and others (1983). The conventions defining redox state are as follows: (1) an operational valence of zero is assigned to O and H in aqueous species and phases; (2) an operational valence of -2 is assigned to H_2 (aqueous, (aq)) and +4 for O_2 (aq); (3) the formal valence is used for any element or species that can change oxidation state in the chemical system under consideration; (4) an operational valence of zero is assigned to any element or species that does not change oxidation state in the system; (5) the operational valence of phases and aqueous complexes is computed as the sum of the products of the operational valence and the stoichiometric coefficient in the phase or aqueous species; and (6) the redox state of an aqueous solution, RS, is the summation of only the element constraints selected in the model. These redox conventions simplify the accounting of electron transfer in reactions. Examples of operational valence of selected species and phases are--

Species	Operational Valence, v_i	Phase	Operational Valence, v_i
Ca^{2+}	0.0	$CaCO_3$	4.0
Mg^{2+}	0.0	$CaMg(CO_3)_2$	8.0
Na^+	0.0	$Ca_{1.05}Mg_{0.9}Fe_{0.05}(CO_3)_2$	8.1
K^+	0.0	$NaCl$	0.0
Fe^{2+}	2.0	$CaSO_4$	6.0
Fe^{3+}	3.0	$FeOOH$	3.0
Cl^-	0.0	Fe_2O_3	6.0
$SO_4{}^{2-}$	6.0	$FeCO_3$	6.0
$H_2S_{(aq)}$	-2.0	FeS	0.0
HS^-	-2.0	FeS_2	0.0
$HCO_3{}^-$	4.0	$CaSO_4 \cdot 2H_2O$	6.0
$CH_{4(aq)}$	-4.0	"CH_2O"	0.0
$CaHCO_3{}^+$	4.0	$KAlSi_3O_8$	0.0
$H_2CO_{3(aq)}$	4.0	$Al(OH)_3$	0.0
$N_{2(aq)}$	0.0	KOH	0.0
$NO_3{}^-$	5.0	$H_2S_{(g)}$	-2.0
$NH_4{}^+$	-3.0	$CH_{4(g)}$	-4.0
$Fe(OH)_2{}^+$	3.0	$CO_{2(g)}$	4.0
$FeSO_{4(aq)}$	8.0	$H_{2(g)}$	-2.0
H^+, OH^-	0.0	$O_{2(g)}$	4.0
$H_{2(aq)}$	-2.0	$N_{2(g)}$	0.0
$O_{2(aq)}$	4.0	$NH_{3(g)}$	-3.0
H_2O	0.0	CaF_2	0.0

The above list is obviously incomplete, but should serve as an example for other species and minerals that might be considered. For example, in natural environments, sodium occurs only in the +1 oxidation state. In NETPATH it is normally not necessary to consider electron-transfer reactions involving sodium, so an operational valence of zero is assigned. If, however, the intent was to interpret waters that had evolved from reaction of sodium metal (Na°) with water, an operational valence of -1 would be assigned to the phase Na°.

It is important to re-state rule 6 (above) for defining the redox state of aqueous solutions. In NETPATH the redox state of an aqueous solution, RS, includes only the constraints selected in the model--that is,

$$RS = \sum_{i=1}^{I} m_i v_i \quad ,$$

where m is the molality of the ith species of operational valence v_i, and I is the total number of analyzed aqueous species necessary to define the total elemental concentrations of the selected constraints. The value of RS then depends on the actual selection of constraints in the model. For example, if a water contained dissolved inorganic carbon, sulfate and ferrous iron, and the selected constraints for the model were carbon and sulfur (only), the RS would be computed considering the dissolved inorganic carbon and sulfur, but would not include the contribution from ferrous iron. If the model were subsequently expanded to include iron as a constraint, the RS would automatically be summed for dissolved inorganic carbon, sulfur and iron species in solution.

Because dissolved organic carbon (DOC) can represent the sum of numerous organic species of differing operational valences, DB accepts data for the average redox state of DOC. The default value is zero. The redox state of the DOC can be modified further in NETPATH using <E>dit, and Redox state of DOC (see below). As a general rule, the oxidation state, $\nu_{org.}$ of carbon in organic compounds or species containing carbon, hydrogen, and oxygen is

$$\nu_{org.} = 2\left(\frac{O}{C}\right) - \left(\frac{H}{C}\right)$$

where O/C and H/C are the atomic ratios of oxygen to carbon and hydrogen to carbon in the organic compound or species. For example, the formal oxidation state of carbon in lignite having the average composition, $CH_{0.8}O_{0.2}$ is -0.4 and the formal valence for the lignite molecule is 0.0 (see for example, Stumm and Morgan, 1981, p. 420). But by the previously defined redox conventions, an operational valance of -0.4 is assigned to the lignite molecule in NETPATH.

Total Dissolved Carbon

For each element selected as a constraint, NETPATH considers the total concentration of that element in the aqueous solutions. Even though data are entered separately for selected oxidation states of sulfur, nitrogen, and carbon in DB, total concentrations of each oxidation state of an element are summed to define the total concentration of the element in solution in NETPATH. This definition has some special consequences as regards total dissolved carbon (TDC), which is defined, in millimoles per kilogram H_2O, as the sum of total dissolved inorganic carbon (TDIC), dissolved methane, and dissolved organic carbon (DOC):

$$m_{TDC} = m_{TDIC} + m_{CH_4} + m_{DOC} \quad .$$

In selecting carbon as a constraint, all mass-balance calculations are constrained by the total dissolved carbon concentrations and the redox states of the aqueous solutions include contributions from all three carbon oxidation states (if appropriate analytical data are given). The definition of TDC allows NETPATH to treat combined reactions involving both inorganic and organic species. Applications include (1) the inorganic carbonate system, if only TDIC is entered in DB, (2) organic carbon systems, if DOC and (or) dissolved methane are entered in DB, and (3) mixed inorganic and organic systems if TDIC and (or) DOC and (or) dissolved methane are entered in DB. It is, therefore, possible to consider in NETPATH the degradation of both natural and anthropogenic organic species in mineral-water systems. If no data are available for DOC and dissolved methane, zero values of their concentrations are assumed in NETPATH and TDC is equal to TDIC. Reactions involving organic compounds included as phases can always be considered regardless of the nature of the original analytical data defining the TDC. However, the user should evaluate the appropriateness of reaction models if potentially important analytical data are missing.

The above definition of total dissolved carbon has further consequences to the definition and interpretation of carbon isotope data, particularly as applied to defining (1) the carbon-13 and carbon-14 content of TDC, (2) carbon isotope fractionation factors which are computed relative to the average isotopic composition of TDC in solution, and (3) the initial ^{14}C content, A_0, of total dissolved carbon used in radiocarbon dating. In NETPATH, DOC represents the sum of the moles (expressed in millimoles of carbon per kilogram H_2O) of all dissolved organic species. It is usually not possible to identify all the individual dissolved organic carbon species that make up DOC. Similarly it is difficult to determine the carbon-13 and carbon-14 content, and RS of all individual organic species in solution. Considering these uncertainties, NETPATH accepts an average carbon-13, carbon-14, and RS content for the total DOC. If it is known, for example, from laboratory experiments that one or several of the dissolved organic species which make up the total DOC are reactive, more realistic models would be obtained using data specific to the reactive species. See Example 7 for a test problem using the full definition of TDC.

Phases

A phase is any mineral or gas that can enter or leave the aqueous solution along the evolutionary path. Selected phases should be known to occur in the system, even if in trace amounts. Reaction modeling can usually be refined by more detailed knowledge of the chemical and isotopic composition of phases in the system. For example, reaction models in a carbonate system constrained by carbon, sulfur, iron and sodium could be refined by knowledge of the amounts of iron, sulfate and sodium substituted in carbonate minerals occurring there. Reaction models could be further checked using data for the carbon-13 composition of carbonates in the system.

In defining the stoichiometry of phases it is important to include its redox state. If omitted, a zero redox state is assumed for the phase. If the redox state of a phase is not zero, it must be defined whenever RS is included as a constraint.

Some of the phases under consideration can realistically only be precipitated or dissolved, but not both along the flow path. For example, organic matter can only dissolve. NETPATH allows phases to be marked for "precipitation only" or "dissolution only". If this is done, only models in which the particular phase precipitates or dissolves will be displayed. Examination of the saturation index data from WATEQFP (listed in the generated file, [*Filename*].OUT) and geochemical intuition are often useful in assigning precipitation only or dissolution only attributes to phases. For example, if the WATEQFP calculations showed that gypsum was undersaturated throughout the system under consideration, there would be little reason to see models requiring gypsum precipitation. Therefore, in this case, it would be meaningful to mark gypsum for dissolution only and the total number of models considered by NETPATH could be reduced.

If a phase is known to always react in the system, it is possible to "force" the phase to be included in every model. This way, more realistic models can be selected for further study and the total number of models may possibly be reduced.

A list of phase stoichiometries is stored in the NETPATH.DAT file and phases can be retrieved from this list in running NETPATH. The NETPATH.DAT file contains two phases that have special meaning. These are "EXCHANGE" and "CO2-CH4". The "EXCHANGE" phase is used to define different $[Ca^{2+}+Mg^{2+}]/Na^{+}$ion exchange models. By adjusting the fraction of Ca^{2+} in the $[Ca^{2+}+Mg^{2+}]/Na^{+}$ exchange considered, the user can determine exactly the modeled exchange taking place.

There are four choices available for exchange: "Computed" calculates the Ca^{2+}/Mg^{2+} ratio in exchange from the molar ratio of calcium to magnesium in the final well. Selecting "50/50" results in equal amounts of calcium and magnesium being assumed in the exchange. "Ca/Na" results in pure calcium/sodium exchange. The last choice, "Var. Ca/Mg", allows the user to enter any fraction of Ca in the exchange. This option can be used in place of "50/50" or "Ca/Na" with the value 0.5 or 1.0, respectively.

Several other exchange reactions are included in NETPATH.DAT and selected in Examples 3-3a and 7 (see Examples and Test Problems). The first of these assumes ferrous iron exchanges for sodium on an exchanger according to the reaction

$$FeX + 2Na^{+} \rightarrow Fe^{2+} + Na_2X \quad .$$

The file NETPATH.DAT defines the above reaction as Fe: +1.0, Na: -2.0, and RS: +2.0; written to release Fe(II) to the aqueous solution. The sign of RS is then determined according to the previously stated conventions. If, alternatively, the reaction were written in reverse, removing Fe(II) from the aqueous solution, the sign of RS and the stoichiometric coefficients would be reversed. The results would be unchanged, however, because the sign of the computed mass transfer for the reaction would also be reversed. The direction the reaction actually proceeds depends on the sign of the computed mass transfer and the direction the reaction was written in assigning the sign of RS and the stoichiometric coefficients. In the above example of Fe(II)/Na exchange, a negative mass transfer for the phase "FeII-Na" (see Example 3-3a) would indicate the reaction proceeded in the reverse of that given above in which Fe(II) was transfered to the exchanger releasing Na^+ to solution.

Example 7 (see Examples and Test Problems) considers the possibility of NH_4^+ exchange for Ca^{2+}. In assigning the stoichiometry, NH_4^+: +2.0, Ca^{2+}: -1.0, and RS: -6.0, the reaction is written as

$$(NH_4)_2X + Ca^{2+} \rightarrow CaX + 2NH_4^{+} \quad ,$$

where RS is again assigned by the previously stated conventions. In this case a positive mass transfer would indicate that NH_4^+ was released from the exchanger to solution and calcium taken up by the exchanger in the mole proportion N:Ca of 2:1. Many other ion exchange reactions can be considered in NETPATH, for which the above should serve as examples.

If mixtures of phases are known to react in fixed proportions, such mixtures can be combined and defined as a single phase in mass-balance models. This allows more phases to be included in the model than are separately allowed for the selected constraints. NETPATH.DAT contains one such phase, a CO_2-CH_4 gas mixture, with the added option of defining the fraction of CO_2 in the gas mixture. When the CO_2-CH_4 phase is included and the CO_2 fraction in the mixture defined, all computed mass transfer for this gas mixture always enters or leaves the aqueous solution in the defined CO_2/CH_4 proportion. Incoming and outgassing methane-carbon dioxide gas mixtures can be important in some geochemical environments undergoing methanogenesis. The fraction of CO_2 in the CO_2-CH_4 gas mixture will define the average redox state for the gas mixture. By combining these two gases into one phase, the number of phases under consideration is effectively increased by one. Note that isotopic values for CO_2 and CH_4 are entered, edited, and stored separately. They are combined linearly according to the CO_2/CH_4 ratio for the purpose of isotope computation. Other phase mixtures could be defined in NETPATH.DAT and used in mass-balance models, but there are no general features for editing their mixing proportions without defining a separate phase mixture.

Models

Many more phases can be selected than there are constraints. NETPATH finds every subset of the selected phases that satisfy the chosen constraints. A model is a subset of the selected phases (and the computed mass transfer) that satisfies all the selected constraints. The model is of the form

Initial water + *"Reactant Phases"* → *Final water* + *"Product Phases"* .

In some cases, it becomes necessary to assume that a second water mixing with the initial water, with or without further reaction, results in the observed final water. This additional water is considered a second initial well in the program. The two initial waters are named INIT1 and INIT2. The modeling assumes the two waters mix in some proportion, and then the various reactions of the model take place. The fraction of each well mixed to produce the final water is displayed, in a mixing case. Negative mixing fractions are not allowed (that is, models giving negative mixing fractions are not reported).

Because of the additional variable when considering mixing, one less phase is needed to produce a complete model. Therefore, mixing models will run with one fewer phase than constraints, whereas nonmixing cases require at least as many phases as constraints. If one constraint is not contained in any of the selected phases, a message is displayed at the main screen informing the user that the mixing ratio will be determined by the amount of the constraint in the initial and final wells. No mixing models will be found if more than one constraint is given that is not contained in any phase.

NETPATH also considers the possibility of evaporation or dilution (with pure water). The treatment is similar to mixing, but with the second initial water, INIT2, defined to be pure water. Negative fractions of INIT2 indicate evaporation, and an evaporation factor (≥ 1) is reported for the model. An evaporation factor of 2 would indicate a doubling of the initial water composition by conservative evaporation.

Positive values of INIT2 indicate dilution, and the dilution factor (≥ 1) is reported for the model. A dilution factor of 2 would indicate that the final water represents one half the conservative solute content of the initial water due to dilution with pure water. As with mixing, the total number of phases that can be included in a model is reduced by one when selecting this option. Mixing and Evaporation/Dilution may both be selected for a particular model, in which case the total number of phases that can be included in a model is reduced by two.

Geochemical mass-balance models can be no better than the data on which they are based. Because of a failure to analyze for all dissolved species and (or) analytical error, water analyses rarely are exactly charged balanced. As discussed below, DB offers one option for balancing the charge of the major elements in constructing the .PAT file. Other errors in analytical data, even for charge-balanced water, will be distributed through the calculated mass transfers of all phases containing that element, which will, in tern, likely affect the mass transfers of other phases. Obviously, great care should be taken in selecting reliable chemical, mineralogic, and isotopic data.

The validity of the mass-balance models depends significantly on the geochemical insight of the modeler in selecting appropriate phases in the model. Generally, only phases that occur in the system should be considered in modeling. The results can be improved significantly as more reliable mineral compositional data are used in the modeling, such as the actual anorthite composition of plagioclase feldspars, the iron content of dolomite, the sulfur isotopic composition of sulfate in substitution in calcite, the ^{14}C content of carbon in calcite of the unsaturated zone, and so forth. Geochemical modeling should not be separated from an aggressive effort to study the mineralogy and petrology of the system.

As discussed by Plummer and others (1983) and Plummer (1984), geochemical modeling rarely leads to unique solutions. The modeling process is best suited for eliminating reaction models from further consideration. For example, a mass-balance model could be eliminated if it requires the net precipitation of a phase that is known to be undersaturated in the system. Similarly, a mass-balance model could be eliminated if the predicted isotopic composition of the final water differs significantly from the observed. Although the final results are rarely unique, the elimination process is useful in limiting the number of reaction possibilities.

The effects of hydrodynamic dispersion are not explicitly accounted for in mass-balance modeling. In solving the net mass balance for waters along a flow path, the compositional mixing effects due to hydrodynamic dispersion cannot be separated from the analytical data and become incorporated into the implied phase mass transfer. Although this effect is thought to be negligible in applications to regional flow systems (see Appendix A of Wigley and others, 1978), mass-balance interpretations of localized hydrochemical problems, such as associated with point-source injection, could be in serious error without proper evaluation of the effects of hydrodynamic dispersion. It may be possible to test mass-balance models for uncertainties caused by hydrodynamic dispersion by application of the net mass transfer results in solute-transport models.

NETPATH does not consider the uncertainty in the analytical data. If, for example, a phase is marked for dissolution only, any model requiring precipitation of the phase will not be displayed, even if the amount of the phase precipitated is very small and within the uncertainties of the analytical data. Future developments in mass-balance reaction modeling will hopefully consider uncertainties in the analytical data.

ISOTOPIC CALCULATIONS[2]

NETPATH considers two types of isotopic calculations: (1) isotope mass balance and (2) Rayleigh calculations. Isotope mass balance is included by selecting any of the following as constraints: Carbon-13, C-14 (pmc), Sulfur-34, Deuterium, Oxygen-18, Tritium, and Strontium-87. The user needs to be careful to specify the sulfur, carbon, and (or) strontium stoichiometry and isotopic compositions of appropriate phases in a model. If phases are included that contain C, or S, or Sr, these elements should also be included as constraints. Alternatively, if the isotope is not contained in any phase, any one of the isotope constraints could be used to determine a mixing fraction. In general, Carbon-13, C-14 (pmc), Sulfur-34, and Strontium-87 should be selected as constraints only for processes (reactions) involving the constraint as a source, such as (1) the mixing of two waters, (2) mineral dissolution (without precipitation), or (3) ingassing (without outgassing). An isotope can also be included as a constraint for the special case of precipitation (without dissolution) if, and only if, the fractionation factor, α, is unity. In such a case, the isotopic composition of the precipitate would be identical to that of solution regardless of extent of reaction. Of the isotopes available in NETPATH, this latter case would apply only to precipitation of a strontium-bearing phase from an initial Sr-bearing water, since the $^{87}Sr/^{86}Sr$ fractionation factor, and similarly for other heavy isotopes, is essentially unity.

When there is both a source and sink for a particular isotope in the reaction, regardless of the value of the fractionation factor, it is usually not valid to include the isotope as a constraint in the mass balance, since the problem must be treated as a Rayleigh distillation problem (see for example Wigley, and others, 1978). The Rayleigh calculations are solved in NETPATH using the general case of N non-fractionating inputs and M fractionating outputs considered by Wigley and others (1978, 1979).

After each mass-balance model is calculated, NETPATH computes the $\delta^{13}C$, ^{14}C (pmc), $\delta^{34}S$, and $^{87}Sr/^{86}Sr$ of the final water as a Rayleigh distillation problem for the modeled mass transfer. If $\delta^{13}C$, ^{14}C (pmc), $\delta^{34}S$, and (or) $^{87}Sr/^{86}Sr$ are selected as constraints, the isotopic composition of the final water calculated by the Rayleigh model can be compared with the observed to examine differences between the fractionating differential problem of isotopic evolution and the mass-balance result. For the special valid cases mentioned above where isotopic data are correctly treated as isotope mass-balance problems, the final modeled (using the Rayleigh calculations) isotopic composition will always equal the final observed value. But for the general case of isotope evolution, the calculations usually involve comparing sensitivity of isotopic values computed at the final well to uncertainties in isotopic and compositional data of the selected phases.

---

[2] Throughout this report the abundances of isotopes of carbon-13 and sulfur-34 are given in delta notation, denoted δ, and expressed in units of parts per thousand (per mil, $^{o}/_{oo}$). The delta value is expressed as

$$\delta_i = \left(\frac{R_i}{R_{Std.}} - 1 \right) 1000 \quad ,$$

where R is the $^{13}C/^{12}C$, or $^{34}S/^{32}S$ ratio in the ith species or phase, or in the standard, denoted "Std.". Carbon-14 is expressed in percent modern carbon (pmc). Strontium-87 abundance is expressed as the ratio, $^{87}Sr/^{86}Sr$.

The user should be aware of some of the assumptions in applying the Wigley and others (1978, 1979) Rayleigh distillation equations to net mass-balance results in isotope evolution problems. The equations of Wigley and others (1978, 1979) are analytical solutions to the general differential problem of (carbon) mass balance and isotope mass balance with fractionating output. The two basic equations solved are:

Carbon mass balance

$$d(mC) = \sum_{i=1}^{N} dI_i - \sum_{i=1}^{M} dO_i$$

Isotope mass balance

$$d(RmC) = \sum_{i=1}^{N} R_i^{*} dI_i - \sum_{i=1}^{M} R_i \alpha_{is} dO_i \quad ,$$

where R refers to the isotope ratio, mC is the total concentration of the element, I and O refer to incoming and outgoing masses of the element, such as through dissolution and precipitation, N and M are the total number of incoming and outgoing phases, the superscript * refers to the incoming phases, and α_{is} is the fractionation factor between the ith phase and the solution. The analytical solution to the isotope evolution problem assumes constant relative rates of reaction along the flow path. Thus the ratio of incoming to outgoing mass of an element is assumed constant along the flow path, and equal to that calculated for the net mass-balance problem. Though the net mass transfer can be determined by NETPATH, the relative rates of reactions may vary along the flow path. It is also assumed that a single value of the additive fractionation factor (relative to the solution) applies over the entire length of the flow path. Test calculations (Wigley and others, 1978; Plummer and others, 1983) have shown that, in selected carbonate cases considered, the modeled isotopic outcome is not usually sensitive to uncertainties in relative rates of reaction and variations in additive fractionation factors. This conclusion is tentative and may not be valid for yet untested reactions. One means of testing the validity of isotopic-evolution results was demonstrated by Plummer and others (1983) in simulating the final isotopic composition of a net mass-balance model over a wide range of possible reaction paths. Such calculations require use of a forward reaction simulation code such as PHREEQE[3] (Parkhurst and others, 1980).

Fractionation Factors for the Inorganic Carbon-13 System

Two sets of fractionation factors for the inorganic carbon-13 system are available in NETPATH through the Edit menu. The default set, identified as Mook, selects equilibrium fractionation factors for carbonate phases from Thode and others (1965), Mook and others (1974), and Mook (1980). The second set of carbonate fractionation factors is identified as Deines, and taken from Deines and others, (1974). The Mook set is given as additive fractionation factors, $\epsilon_{i\text{-}j}$, (in per mil), and the Deines and others set is given as the fractionation factor $\alpha_{i\text{-}j}$, as in the original sources.

The fractionation factor, $\alpha_{i\text{-}j}$, between the ith and jth species (phases) is defined as

---

[3] PH-REdox-EQuilibrium-Equations

$$\alpha_{i-j} = \frac{R_i}{R_j} = \frac{1000 + \delta_i}{1000 + \delta_j} \quad ,$$

where R is the isotope ratio, and δ is the isotopic composition in per mil relative to a standard. That is,

$$\delta_i = \left(\frac{R_i}{R_{std.}} - 1 \right) 1000$$

where R_i and $R_{std.}$ are, for carbon-13, the ratio $^{13}C/^{12}C$ in species (phase) i and in the standard.

The additive fractionation factor, ϵ, is related to the fractionation factor, α, by the equation

$$\epsilon_{i-j} = 1000(\alpha_{i-j} - 1) \quad .$$

Friedman and O'Neil (1977) give an extensive review of literature values of α. The fractionation factors of the Mook and Deines sets are from various sources. Generally, they are based on experimental data between 0 and 50 °C (degrees Celsius), but can be applied to temperatures approaching 100 °C. The two sets of fractionation factors are as follows--

Mook (1980)

$$\epsilon_{CO_{2aq}-HCO_3^-} = 24.12 - \frac{9866}{T} \approx \delta^{13}C_{CO_{2aq}} - \delta^{13}C_{HCO_3^-}$$

$$\epsilon_{CO_3^{2-}-HCO_3^-} = 1.63 - \frac{604}{T} \approx \delta^{13}C_{CO_3^{2-}} - \delta^{13}C_{HCO_3^-}$$

$$\epsilon_{calcite-HCO_3^-} = 15.10 - \frac{4232}{T} \approx \delta^{13}C_{calcite} - \delta^{13}C_{HCO_3^-}$$

and

$$\epsilon_{CO_{2gas}-HCO_3^-} = 23.89 - \frac{9483}{T} \approx \delta^{13}C_{CO_{2gas}} - \delta^{13}C_{HCO_3^-} \quad ,$$

where T is temperature in kelvins.

Deines and others (1974)

$$1000\ln\alpha_{CO_{2aq}-CO_2(g)} = -0.91 + \frac{6300.}{T^2}$$

$$1000\ln\alpha_{CO_3^{2-}-CO_2(g)} = -3.4 + \frac{870000.}{T^2}$$

$$1000\ln\alpha_{calcite-CO_2(g)} = -3.63 + \frac{1194000.}{T^2}$$

and

$$1000\ln\alpha_{HCO_3^{-}-CO_2(g)} = -4.54 + \frac{1099000}{T^2} \quad ,$$

where T is temperature in kelvins. Within NETPATH the $\alpha_{i\text{-}CO2(g)}$ values from Deines and others (1974) are converted to $\epsilon_{i\text{-}HCO3}$. The most significant difference between the Mook and Deines and others sets of fractionation factors is the calcite-HCO_3^- fractionation, where, at 25 °C the Deines and others (1974) value is 1.98 per mil and the Mook (1980) value is 0.91 per mil. The CO_3^{2-}-HCO_3^- fractionation factors also differ by about 1 per mil between the two sets, but the difference has little effect in most groundwaters because HCO_3^- is usually the predominant inorganic carbon species.

Additive Fractionation Factors, ϵ, Relative to the Average Isotopic Composition of the Solution

In NETPATH, all fractionation factors are defined relative to the average isotopic composition of the aqueous solution, rather than to an individual aqueous species. Wigley and others (1978, 1979) show that additive fractionation factors for calcite and CO_2(g) relative to the average isotopic composition of the dissolved inorganic carbon in solution are functions of temperature and pH.

As discussed earlier, NETPATH defines total dissolved carbon as the sum of dissolved inorganic carbon, methane, and dissolved organic carbon--that is,

$$m_{TDC} = m_{TDIC} + m_{CH_4} + m_{DOC} \quad ,$$

where m is millimoles per kilogram H_2O, and the subscripts TDC, TDIC, CH_4, and DOC refer to total dissolved carbon, total dissolved inorganic carbon, dissolved methane, and dissolved organic carbon, respectively. According to this definition, the average $\delta^{13}C$ isotopic composition of TDC is

$$\delta^{13}C_{TDC} = \frac{m_{TDIC}\delta^{13}C_{TDIC} + m_{CH_4}\delta^{13}C_{CH_4} + m_{DOC}\delta^{13}C_{DOC}}{m_{TDIC} + m_{CH_4} + m_{DOC}} \quad .$$

The fractionation factors, α, for carbonates calculated relative to the average $\delta^{13}C$ isotopic composition of the aqueous solution depend on (1) the distribution of carbonate species computed by WATEQFP, (2) the $\delta^{13}C$ of TDIC, (3) the mmol/kg H_2O of methane and mmol/kg H_2O of carbon from DOC, and (4) the $\delta^{13}C$ composition of dissolved methane and DOC which is user defined within NETPATH (see the <E>dit, Isotope Data option). The default values of the equilibrium carbon isotope fractionation factors for calcite, and CO_2 gas are calculated relative to the average isotopic composition of the solution using the formal relations:

$$\alpha_{Cal.-Soln.} = \frac{\alpha_{Cal.-HCO_3^-}}{N_{CO_2aq}\alpha_{CO_2aq-HCO_3} + N_{HCO_3} + N_{CO_3}\alpha_{CO_3-HCO_3} + N_{CH_4}\alpha_{CH_4-HCO_3} + N_{DOC}\alpha_{DOC-HCO_3}}$$

and

$$\alpha_{CO_2(g)-Soln.} = \frac{\alpha_{CO_2(g)-HCO_3^-}}{N_{CO_2aq}\alpha_{CO_2aq-HCO_3} + N_{HCO_3} + N_{CO_3}\alpha_{CO_3-HCO_3} + N_{CH_4}\alpha_{CH_4-HCO_3} + N_{DOC}\alpha_{DOC-HCO_3}}$$

where N_i is the mole fraction of the subscripted carbon species (relative to TDC), and $\alpha_{CH4-HCO3}$ and $\alpha_{DOC-HCO3}$ are treated as kinetic isotope fractionation factors and defined from the analytical data, if available--that is,

$$\alpha_{CH_4-HCO_3} = \frac{1000 + \delta^{13}C_{CH_4}}{1000 + \delta^{13}C_{HCO_3}}$$

$$\alpha_{DOC-HCO_3} = \frac{1000 + \delta^{13}C_{DOC}}{1000 + \delta^{13}C_{HCO_3}}$$

and

$$\delta^{13}C_{HCO_3} = \frac{1000 + \delta^{13}C_{TDIC}}{N_{CO_2aq}\alpha_{CO_2aq-HCO_3} + N_{HCO_3} + N_{CO_3}\alpha_{CO_3-HCO_3}} - 1000 \quad .$$

The default equilibrium carbon isotope fractionation factors for all carbonate minerals are initially defined as that for calcite. The section below on <E>dit, Isotope Data shows how user-defined fractionation factors can be entered into NETPATH. The default equilibrium fractionation factor for methane gas relative to solution is undefined and must be entered by the user.

Inspection of the above equations shows that if dissolved methane and DOC have zero concentrations in the .PAT file, all calculations in the carbon system of NETPATH reduce to the usual definitions for the inorganic carbon system. The user should be aware of the consequences of entering data for DOC and (or) dissolved methane in DB, as these data fully impact the definition of total dissolved carbon, the redox state of the solution, and the treatment of the isotopic evolution of the carbon system. In running isotope-evolution problems in NETPATH with analytical data that include DOC and (or) dissolved methane, NETPATH prints the modeled $\delta^{13}C$ of both the dissolved inorganic carbon (for direct comparison with known analytical data) and the $\delta^{13}C$ of total dissolved carbon (which is usually not measured directly, but can be computed from the analytical data). Comparison of modeled $\delta^{13}C$ isotopic compositions with analytical data at the final well is of considerable value in testing model sensitivity to uncertainties in data and, in some cases, eliminating models from further consideration.

If the carbon-14 content is also modeled in fractionating processes, NETPATH initially assigns default values of the additive fractionation factors for carbon-14 as twice those for carbon-13 (Craig, 1954). Alternatively, the user has the option of editing all fractionation factors used by NETPATH through the <E>dit, Isotope data screens (see later discussion).

Sulfur-Isotope Fractionation Factors

As with the carbonate phases, the fractionation factor for sulfur-bearing phases is defined relative to the average isotopic composition of sulfur in the solution. However, NETPATH does not include dissolved organic sulfur. The average isotopic composition of sulfur in solution is calculated in NETPATH using the total concentrations of SO_4 and H_2S and their individual isotopic values specified in DB--that is,

$$\delta^{34}S_T = \frac{m_{SO_{4T}}\delta^{34}S_{SO_{4T}} + m_{H_2S_T}\delta^{34}S_{H_2S_T}}{m_{SO_{4T}} + m_{H_2S_T}} \quad .$$

Only one default isotope fractionation factor is calculated for the sulfur system. This applies to precipitation of sulfide phases from solution, and is specifically intended to describe kinetic, microbial fractionation of sulfur accompanying sulfate reduction and precipitation of iron sulfide phases. It is initially assumed that the sulfur isotopic composition of sulfide phases is that of the dissolved hydrogen sulfide in solution. The additive fractionation factor for sulfur-34 between, for example, pyrite (pyr.) and solution (soln.) is then

$$\epsilon_{pyr.-soln.} = 1000(\alpha_{pyr.-soln.} - 1) = \delta^{34}S_{H_2S_T} + \epsilon_{pyr.-H_2S} - \delta^{34}S_T \quad .$$

The additive fractionation factor between pyrite and hydrogen sulfide, $\epsilon_{pyr.-H_2S}$, is initially assumed to be zero in NETPATH, but may be changed by selecting the appropriate phase number on the fractionation factor screen. Several calculations are possible in defining the default sulfur isotope fractionation factor depending on the available data in DB: (1) If both sulfate and sulfide and their isotopic values are defined for the water analysis in DB, the default sulfur isotope fractionation factor for sulfide phases will be calculated directly from the analyzed data as given above; (2) if no value of the sulfur isotopic composition of dissolved hydrogen sulfide is available, the correlation introduced by Plummer and others (1990) is used to estimate $\delta^{34}S_{H2S}$ based on the observed sulfur isotopic composition of dissolved sulfate and water temperature--

$$\delta^{34}S_{H_2S} = \delta^{34}S_{SO_4} - 54. + 0.40t \quad ,$$

where t is water temperature in °C; and, (3) if no data are available for the sulfur isotopic composition of dissolved sulfate, the default fractionation factor is undefined. The default fractionation factors may be changed by selecting the appropriate phase number appearing on the screen displaying fractionation factors.

The additive fractionation factor for a sulfate-bearing phase, such as gypsum (gyp.), is also defined relative to the average isotopic composition of sulfur in solution,

$$\epsilon_{gyp.-soln.} = \delta^{34}S_{SO_4} + \epsilon_{gyp.-SO_4} - \delta^{34}S_T \quad .$$

The default additive fractionation factor between gypsum and sulfate, $\epsilon_{gyp.-SO_4}$, is defined to be zero, and may be edited by selecting the appropriate phase number under <E>dit, Isotope Data (see below). Inspection of the above equations shows that if no hydrogen sulfide is present in solution, the sulfur-system fractionation factors are the same as for the sulfate system. In deriving the above fractionation factors for sulfur-bearing phases relative to the average isotopic composition of sulfur in solution, use is made of the approximation, $\epsilon_{i\text{-}j} \approx \delta_i - \delta_j$.

RADIOCARBON DATING

In application to radiocarbon dating, the mass transfer calculated by NETPATH is used to adjust the initial ^{14}C composition, A_0, for all sources and sinks of carbon which affect the carbon mass transfer between initial and final wells along the flow path. This procedure calculates the ^{14}C composition at the final well, adjusted for chemical reaction but not radioactive decay, denoted A_{nd}. Radiocarbon dating is then applied to the final well on the flow path using A_{nd} and the observed value, A.

The reaction path between the initial and final waters may be open or closed to carbon phases such as CO_2 gas, organic matter, and carbonate minerals. All carbon-mass transfer found to leave the aqueous solution between the initial and final waters, such as through carbonate mineral precipitation, or outgassing of CO_2 or methane gas (negative mass-transfer coefficients) is assumed to leave by a Rayleigh-distillation process which uses the previously defined isotopic fractionation factors. All carbon-mass transfer computed for minerals or gases that enter the aqueous phase between the initial and final waters (positive mass-transfer coefficients) are assumed to enter the aqueous solution without isotopic fractionation, and have $\delta^{13}C$ and ^{14}C (pmc) compositions as defined for the phases (see <E>dit, Isotopic data).

The initial water may represent any point on the flow path. For example, the initial water may be located in the recharge zone where it may be in exchange equilibrium with the soil atmosphere; or it could be just downgradient of the recharge zone and isolated from the soil atmosphere; or it could be a water farther downgradient of the recharge zone, but still upgradient of the final water. Definition of the initial ^{14}C value in NETPATH depends on the segment of the flow system being dated. For example, in calculating the travel time between two deep confined wells, the initial ^{14}C value might be defined as the measured ^{14}C composition of the upgradient well. The modeled ^{14}C "age" of the final water would, in this case, represent only the travel time between the initial and final well. To find the actual age of the final water, it would be necessary to add to this travel time the actual age of the initial water. The age of the initial water is the sum of the residence time of the water in the recharge zone before isolation from the modern reservoir, and the travel time of the water from the first point of isolation from the modern reservoir to the initial well. It is not possible to use ^{14}C data to calculate the residence time of the water in the open-system recharge zone because the carbon isotopic composition of this zone is continually buffered by the modern soil reservoir.

Radiocarbon dating then begins at the point in the aquifer where the water is first isolated from the soil reservoir (see for example Wigley and others, 1978) and here the initial ^{14}C composition is referred to as A_0. NETPATH considers nine options (models) for definition of A_0 or the initial ^{14}C content along the flow path (see below). Caution needs to be exercised in using these options because each model has been developed for specific reaction conditions. It may not be appropriate, or desirable to apply these models to waters that have experienced extensive chemical evolution at points downgradient from the recharge environment.

If appropriate ^{14}C data are available, NETPATH will attempt radiocarbon dating of the final water. The age-dating procedure depends on three values of ^{14}C activity: (1) the initial ^{14}C value for the initial well, A_0 (such as the estimated prenuclear-detonation ^{14}C content of the recharge water at the moment the water became isolated from a modern source); (2) the adjusted ^{14}C value calculated at the final well by accounting for reaction effects to the initial ^{14}C, A_{nd}; and (3) the measured ^{14}C content in the final water, A, entered in DB. In ^{14}C age-dating, NETPATH computes values of A_{nd} for each reaction model using the defined value of A_0, defined ^{14}C isotopic content of carbon sources, defined ^{14}C fractionation factors, and the computed carbon-mass transfer. The calculated A_{nd} is displayed and used to calculate an adjusted ^{14}C age, according to the equation:

$$\Delta t(years) = \frac{5730}{\ln 2}\ln\left(\frac{A_{nd}}{A}\right) \quad .$$

The ^{14}C age is the travel time, in years, between the initial and final well.

This approach to radiocarbon dating is more generalized than that of Reardon and Fritz (1978) and Fontes and Garnier (1979) because an unlimited number of reaction possibilities can be considered. The NETPATH modeling approach to radiocarbon dating also differs conceptually from that of Cheng and Long (1984) who treated the reaction corrections as a forward simulation in the subroutine CSOTOP adapted to the reaction simulation code PHREEQE (Parkhurst and others, 1980). NETPATH uses the inverse modeling approach (Plummer, 1984). Each model found by NETPATH is constrained by the analytical data and if treated in a forward simulation, the mass transfer results would reproduce identically the final water composition. PHREEQE simulations assume arbitrary reaction models and are not constrained by the analytical data at the end-point. Forward simulations such as computed with PHREEQE-CSOTOP are useful in investigating possible trends in isotope evolution in response to hypothetical reactions. In complex hydrochemical systems it is very difficult to use forward simulation methods to find reaction models that reproduce the final water chemistry. When analytical data are available at the initial and final points along a flow path, the inverse modeling approach, such as used by NETPATH, will find all possible reaction models consistent with the available data.

Initial ^{14}C Activity Models

Several models have been proposed in the hydrochemical literature for estimation of A_0. NETPATH considers 9 possible means of defining the initial ^{14}C. These are termed: (1) original data, (2) mass balance, (3) Vogel, (4) Tamers, (5) Ingerson and Pearson, (6) Mook, (7) Fontes and Garnier, (8) Eichinger, and (9) user-defined. Several of these cases (3-7) are summarized and evaluated by Fontes and Garnier (1979).

For consistency with the definition of total dissolved carbon in NETPATH (see above), it is necessary to modify the values of A_0 calculated from the literature models which consider only the dissolved inorganic carbon. The modification is as follows:

$$A_{oTDC} = \frac{A_{oTDIC}m_{TDIC} + {}^{14}C_{CH_4}m_{CH_4} + {}^{14}C_{DOC}m_{DOC}}{m_{TDC}} \quad ,$$

where the subscripts TDC, TDIC, and DOC refer to total dissolved carbon, total dissolved inorganic carbon, and dissolved organic carbon, respectively, and m is molal concentration of the subscripted quantity in the initial water. If DOC and dissolved methane concentrations are zero in the initial water, the initial ^{14}C is defined identically to that for the inorganic carbon system. In calculating A_{oTDC}, $A_{o(TDIC)}$ is first calculated considering only the inorganic carbon system and using fractionation factors defined for the inorganic carbon system, as in the original references and (or) as summarized in Fontes and Garnier (1979). These values of $A_{o(TDIC)}$ are then adjusted for the DOC and dissolved methane (if present) according to the above equation. If DOC and (or) dissolved methane are entered in DB for the initial water and radiocarbon age-dating is attempted, their ^{14}C contents must be specified (see <E>dit Isotope Data) for proper definition of A_{oTDC}.

NETPATH lists the various A_0 models, allows selection of a model, and editing of data used in the $A_{0(TDIC)}$ model. An example of the $A_{0(TDC)}$ is as follows:

```
Initial Carbon-14, AO, (percent modern)
       for Total Dissolved Carbon
       Model                Initial Well
 1 : Original Data        :     33.05
 2 : Mass Balance         :     52.33
 3 : Vogel                :     85.00
 4 : Tamers               :     53.46
 5 : Ingerson and Pearson:      52.33
 6 : Mook                 :     53.80
 7 : Fontes and Garnier   :     46.52
 8 : Eichinger            :     47.57
 9 : User-defined         :    100.00
Enter number of model to use (<Enter> for 'Mass Balance')
```

Note that the initial ^{14}C value for the Vogel model is given in the above example as 85.00. Because this is the value proposed by Vogel for the inorganic carbon system, it can be assumed that in the above example the concentrations of DOC and dissolved methane were zero in the initial water, or if present, had an average Carbon-14 content of 85 pmc.

For each case of defining $A_{0(TDIC)}$ (in percent modern carbon (pmc)), NETPATH has pre-defined default values of parameters associated with the model. Once a particular case for definition of A_0 is selected, opportunity is given for altering the default values used in the particular case, or if desired, default values for all A_0 models may be edited.

Case (1) uses as a value of $A_{0(TDIC)}$, the ^{14}C content of dissolved inorganic carbon defined for the initial water in DB. The "mass balance" model, case (2), was used by Plummer and others (1990) and is similar to Tamers model (Tamers, 1967, 1975; Tamers and Scharpenseel, 1970). A chemical mass balance is made on the initial water composition assuming reaction of pure water with calcite, dolomite, gypsum and CO_2 gas. The default ^{14}C content of the CO_2 gas is assumed to be 100-percent modern, and that of carbonate sources 0-percent modern. Case (3) refers to the Vogel model (Vogel, 1967; 1970; and Vogel and Ehhalt, 1963) which assigns $A_{0(TDIC)}$ to be 85 percent modern. The "Tamers model", case (4), is similar to the "Mass Balance" model described above, except that the mass balance is performed only on carbonates and CO_2 gas. The default ^{14}C values are carbonates, 0 percent modern, and CO_2, 100 percent modern.

The model of Ingerson and Pearson (1964), case (5), assumes a carbonate dissolution model to estimate $A_{0(TDIC)}$ based on ^{13}C data for the inorganic carbon system (for further discussion see Wigley and others, 1978; Fontes and Garnier, 1979). The initial ^{14}C content is estimated according to

$$A_{oTDIC} = \frac{\delta^{13}C - \delta^{13}C_c}{\delta^{13}C_{CO_2} - \delta^{13}C_c}(A_{CO_2} - A_c) + A_c \quad ,$$

where $\delta^{13}C$ is the ^{13}C composition of total dissolved inorganic carbon in the initial water, $\delta^{13}C_c$ is the ^{13}C content of dissolving carbonate (default 0 per mil), $\delta^{13}C_{CO2}$ is the ^{13}C content of soil gas CO_2 (default -25 per mil), A_{CO2} is the ^{14}C content of soil gas CO_2 (default 100 pmc), and A_c is the ^{14}C content of soil carbonate minerals (default 0 pmc).

More complex models for the initial ^{14}C content, A_0, of total dissolved inorganic carbon have been proposed by Mook (1972, 1976, 1980), Fontes and Garnier (1979), and Eichinger (1983). All assume that carbon isotopic equilibrium occurs in one or more steps in the evolution of the recharge water. Mook assumes that all carbonate minerals, gas, and aqueous species are both in chemical and in isotopic equilibrium with the soil CO_2 gas in defining $A_{o(TDIC)}$. The Mook model considers the system to be initially open to soil CO_2. Accordingly, Mook obtains from a chemical mass balance (Fontes and Garnier, 1979)

$$
\begin{aligned}
A_{oTDIC} = {} & N_{CO_{2aq}} A_{CO_{2aq(0)}} + 0.5(1 - N_{CO_{2aq}})(A_{CO_{2aq(0)}} + A_{ls(0)}) \\
& + \left(A_{CO_{2gas(0)}} \left(1 - 2x10^{-3} \epsilon_{CO_{2gas} - HCO_3^-} \right) - 0.5(A_{CO_{2aq(0)}} + A_{ls(0)}) \right) \\
& \cdot \frac{\delta^{13}C - N_{CO_{2aq}} \delta^{13}C_{CO_{2aq(0)}} - 0.5(1 - N_{CO_{2aq}})(\delta^{13}C_{CO_{2aq(0)}} + \delta^{13}C_{ls(0)})}{\delta_{CO_{2gas(0)}} - \epsilon_{CO_{2gas} - HCO_3^-}(1 + 10^{-3}\delta^{13}C_{CO_{2gas(0)}}) - 0.5(\delta^{13}C_{CO_{2aq(0)}} + \delta^{13}C_{ls(0)})} ,
\end{aligned}
$$

where N refers to the fraction of the total dissolved inorganic carbon represented by the subscripted species, the subscript "(0)" refers to the initial condition before isotope exchange reaction, and the subscript "ls" refers to the limestone. See Mook (1972, 1976, 1980) for further details.

Fontes and Garnier (1979) consider a two-stage evolution of recharge waters accounting for dissolution and isotopic exchange of carbonate minerals with CO_2 in the unsaturated zone and isotopic exchange with the carbonate rocks in the saturated zone. A chemical mass balance is performed similar to that of Tamers, with provision for base exchange, to define the mass of carbon dissolved from inorganic sources, C_M. Like the Mook model, the Fontes and Garnier model reduces to Tamer's, if there is no isotopic exchange. Fontes and Garnier (1979) obtain the following relation for $A_{o(TDIC)}$

$$
\begin{aligned}
A_{oTDIC} = {} & \left(1 - \frac{C_M}{C_T} \right) A_{CO_{2gas}} + \frac{C_M}{C_T} A_M + (A_{CO_{2gas}} - 0.2\epsilon_{CO_{2gas} - calcite} - A_M) \\
& \cdot \frac{\delta^{13}C - (C_M/C_T)\delta^{13}C_M - (1 - (C_M/C_T))\delta^{13}C_{CO_{2gas}}}{\delta^{13}C_{CO_{2gas}} - \epsilon_{CO_{2gas} - calcite} - \delta^{13}C_M} ,
\end{aligned}
$$

where C_T is the total millimoles of dissolved inorganic carbon in the recharge water with ^{13}C content of $\delta^{13}C$, A refers to the ^{14}C content of the subscripted carbonate minerals, M, and soil CO_2 gas, and ϵ refers to the additive fractionation factor between CO_2 gas and calcite,

$$
\epsilon_{CO_{2gas} - calcite} = -12.38 + 0.10t \quad ,
$$

as given by Fontes and Garnier (1979), where t is temperature in °C. This fractionation factor is based on the data of Deines and others (1974). Fontes and Garnier (1979) discuss some of the possible limitations of their model.

Eichinger (1983) developed an isotope-exchange/mass-balance model accounting for equilibrium isotopic exchange for introduction of soil CO_2 into the water and equilibrium exchange between dissolved inorganic carbon and the carbonate rock. The model is similar to that of Ingerson and Pearson with modification for equilibrium isotopic exchange. Eichinger's equation for $A_{o(TDIC)}$ is

$$A_{oTDIC} = \frac{\delta^{13}C - \delta^{13}C_{s,eq}}{\delta^{13}C_{b,ex} - \delta^{13}C_{s,eq}} \cdot \left(N_{CO_{2aq}} + 0.5N_{HCO_3^-}\right)A_{CO_{2gas}} \quad ,$$

where

$$\delta^{13}C_{b,ex} = N_{CO_{2aq}}\left(\delta^{13}C_{CO_{2gas}} - \epsilon_{CO_{2gas}-CO_{2aq}}\right) + 0.5N_{HCO_3^-}\left(\delta^{13}C_{CO_{2gas}} - \epsilon_{CO_{2gas}-CO_{2aq}} + \delta^{13}C_s\right)$$

and

$$\delta^{13}C_{s,eq} = N_{CO_{2aq}}\left(\epsilon_{CO_{2aq}-calcite}\right) + N_{HCO_3^-}\left(\epsilon_{HCO_3^--calcite}\right) + \delta^{13}C_s \quad .$$

In the Eichinger model, the subscript "s" refers to the solid, the subscript "s, eq" refers to the solid at isotopic equilibrium with the dissolved inorganic carbon of the solution, and the subscript "b, ex" refers to the state before isotope exchange. N refers to the fraction of total inorganic carbon as CO_2 (aq) and as HCO_3^-. The equation for $\delta^{13}C_{s,eq}$ has been corrected for a misprint in the original Eqn. (6) of Eichinger (1983), (Fontes, J.-Ch., written communication, 1991).

Carbon-13 Composition of Soil Gas Carbon Dioxide

The Models of Ingerson and Pearson, Mook, Fontes and Garnier, and Eichinger all require definition of $\delta^{13}C$ of soil gas CO_2. In the absence of measurement, NETPATH allows four options in defining this value. They are--

```
Choices for delta C-13 (per mil) in soil gas CO2
 0 : User-defined Value
 1 : Mass Balance - no fractionation
 2 : Mass Balance - with fractionation
 3 : Open System (gas-solution equilibrium)
```

Choice 0 is simply user-defined. Choices 1 and 2 each perform an isotope mass balance assuming the initial water evolved by reaction of soil CO_2 with calcite, dolomite and gypsum according to the reaction

$$CO_2(g) + H_2O + CaCO_3 + CaMg(CO_3)_2 + CaSO_4 = Initial\ Water \quad .$$

An isotope mass balance defines the ^{13}C content of the CO_2 gas that entered the solution prior to reaction with the carbonate, $\delta^{13}C_{CO2aq(0)}$,

$$\delta^{13}C_{CO_{2aq(0)}} = \frac{\delta^{13}C_{TDIC}C_{TDIC} - \Delta_{calcite}\delta^{13}C_{calcite} - 2\Delta_{dolomite}\delta^{13}C_{dolomite}}{\Delta_{CO_{2gas}}} ,$$

where Δ refers to the mmoles of calcite, dolomite, or CO_2 dissolved in forming the initial water, and C_{TDIC} is the total concentration of inorganic carbon in the initial water (mmoles). Option (1) assumes that the $CO_{2(aq(0))}$ entered without fractionation, such as from complete injection of soil gas CO_2, and therefore the ^{13}C content of the soil gas is defined directly from the mass balance. Option (2) assumes that the $CO_{2(aq(0))}$ dissolved into the initial water maintaining isotopic equilibrium with the soil gas prior to dissolution of calcite, dolomite and gypsum, that is, gas exchange is assumed to be rapid relative to mineral dissolution. The $\delta^{13}C$ of soil gas is then calculated for option (2) from the $\delta^{13}C$ of $CO_{2(aq(0))}$ calculated from the mass balance and the equilibrium isotopic fractionation between $CO_{2(aq)}$ and $CO_{2(g)}$--that is, for option (2),

$$\delta^{13}C_{CO_{2gas}} = \delta^{13}C_{CO_{2aq(0)}} + \epsilon_{CO_{2g}-CO_{2aq}} .$$

For option (2), it is assumed that calcite, dolomite, and gypsum subsequently dissolve under closed system conditions. Option (3) assumes a completely open system in which $\delta^{13}C$ of soil gas is defined from $\delta^{13}C$ of the initial water and the $CO_{2(g\text{-}soln)}$ equilibrium fractionation defined earlier.

Isotopic Exchange

In addition to allowing phases to be marked for dissolution only, or precipitation only, NETPATH allows a phase to be considered to dissolve and precipitate in a single specified amount, a process similar to isotopic exchange. In NETPATH isotope exchange is analogous to the process of recrystallization or ripening of a pure phase. Although this does not affect the mass balance, isotopic computations will be affected. Isotope exchange may be necessary to model observed isotopic values without disturbing a realistic mass-balance model. During isotopic exchange, the indicated mass of the solid is dissolved at its defined isotopic value and precipitated with an equilibrium fractionation step using the Rayleigh-distillation equations. In the case of carbonates, extensive isotopic exchange can have a profound effect on the modeled ^{13}C and ^{14}C content. This option should not be selected without valid evidence for the occurrence of isotopic exchange in the system. Any phase marked for isotopic exchange may be included in the model and can have net mass transfer (precipitation or dissolution) in addition to the specified amount (in millimoles per kilogram H_2O) exchanged per kilogram of water. The amount exchanged is also displayed if the phase has zero net mass transfer but a specified amount of isotopic exchange. See Example 5 (below) for further information.

RUNNING DB

The first step in using NETPATH is to enter the data into a database. DB is a simple database program that accepts data in various units, allows editing of the data, stores the data in a .LON file and produces a modified WATEQF (Truesdell and Jones, 1974; Plummer and others, 1976) input file which may then be converted into a .PAT file for use in NETPATH.

Entering Data

To start DB, Type: DB [*Filename*], where [*Filename*] is optional on the command line. [*Filename*] is the basic file name to which a suffix will be added for each type of data file produced from the data. If a file name is not entered on the command line, the user is prompted for the file name.

The main screen will now be displayed. If the file contains previously entered data, the names of the wells (or the first 30, if there are more than 30 wells) will be displayed on the screen. DB accepts a total of 50 water analyses per file. An example of the main screen to DB is listed below:

```
 #  Well Name                           | #  Well Name
========================================|=======================================
 1) Nashville #4                        | 2) Glynn Co. Casino
 3) FAA Hilliard                        | 4) Landings #1 (Skidaway Is.)
 5) Fernandina Beach Softball Park      | 6) Metter '75
 7) McRae #3                            | 8) Glynn Co. Recreation (Blythe Is)
 9) Blackshear                          |10) Hazelhurst #3
11) Becker                              |12) Stephen Foster Park
13) Homerville #2                       |14) Alma #3
15) Robinson - KOA                      |16) Twin City #2
17) Richmond Hill                       |18) Savannah #23
19) J. Stanfield                        |20) Savannah Beach (Tybee Is.)
21) Avera #2                            |22) Kent Canning
23) Wadley #2                           |24) Midville Exp Sta TW 1
25) City of Swainsboro #3               |26) Hopeulikit TW 2
27) Town of Cadwell #1 (1976)           |28) DNR Laurens Co. #3
29) Albert S. Mercer (1977)             |30) Laurens Park Mill #3
================================================================================
<A>dd, <D>elete, <E>dit, <M>ove, <N>ext page, <P>rint, <S>ave, or <Q>uit?
```

A list of the available options is displayed at the bottom of this screen. These are accessed by typing the first letter of the desired option. Descriptions of these options follow.

Add

This selection allows data for a water analysis to be entered into the database. After a position is selected where the new analysis should be added, the user is asked whether all data are to be entered or just the data used in NETPATH. Some of the data stored by DB (in the .LON file) do not affect the results of NETPATH modeling, and these need not be entered. This applies to information such as site ID, well depth, well elevation, length of casing, and flow-path designation.

Next, the units to be used for the given water analysis must be selected. DB accepts analytical data in units of millimoles per liter (mmol/L), milliequivalents per liter (meq/L), milligrams per liter (mg/L), parts per million (ppm), or millimoles per kilogram of water (mmol/kg H_2O). If the numbers entered are to be used in NETPATH without conversion, option 4 (mmol/kg H_2O) should be selected. Some of the values associated with the analysis must be entered in specific units. These are: dissolved oxygen, in mg/L as O_2; dissolved methane, in mg/L as CH_4; DOC, in mg/L as C; temperature in °C, density, in grams per cubic centimeter (g/cm^3) (default is 1.0); Eh in volts; tritium, in tritium units (TU); ^{14}C of TDIC in percent modern carbon (pmc); $^{87}Sr/^{86}Sr$ as the ratio; and all other stable-isotope data in per mil.

DB also allows selection of one of two options in calculating individual ion activity coefficients (Extended Debye-Hückel and Davies equations; see Plummer and others, 1976), and one of three options in selecting the way TDIC is to be specified. These are (1) the uncorrected titration alkalinity (as HCO_3^-), (2) the corrected (carbonate) alkalinity (as HCO_3^-), and (3) TDIC. Plummer and others, (1976) give more information on input data to WATEQF.

For each remaining value to be entered, a prompt is displayed which includes the name of the value and, if applicable, the units in which it is to be entered. A special routine is used to get the input from the user. <Enter> causes a missing value condition to be stored for the datum. If an integer value is to be entered, a decimal point is optional in the input.

After all the requested values have been entered, a screen is displayed with the analytical data for the water analysis for the current well. This screen is identical to that obtained with the <E>dit function of the program. See the section on Edit for details. This is an example of the display screen of DB for a particular water analysis:

```
 41) Stephen Foster Park          27E003   304943082214701  85/08/14 @ 10000

   1)Temp.       2)pH          3)diss. O2    4)Alkalinity  5)Tritium
       21.7          7.31          0.0          322.7           0.2
   6)H2S         7)Ca2+        8)Eh          9)Mg2+       10)Na+
        3.4          68.4      SO4/H2S          34.4           59.2
  11)K+         12)Cl-        13)SO42-      14)F-         15)SiO2
        3.6          72.1          61.6          1.0           36.
  16)Br-        17)B          18)Ba2+       19)Li+        20)Sr2+
       0.22         0.060         0.044         0.025         0.620
  21)Fe         22)Mn         23)NO2+NO3-N  24)NH4+kjd-N  25)PO4-P
      0.059         0.002     <   0.010          1.4          0.001
  26)DOC        27)Sp. Cond.  28)Density    29)13C/12C    30)14C
        6.8          786.          1.0         -12.80          1.63
  31)34SSO4     32)34SH2S     33)D          34)18O        35)Diss CH4
       31.5         -30.7          -7.5         -2.15     **********
  36)Sr 87/86   37)Al3+
    0.70849           0.0
Enter # of value to change, 0 to exit :
```

Less than values, denoted " < ", may be stored with DB (see for example NO_3^- above) in the .LON file. In making the .PAT file, less than values are written as zero. Several choices in defining Eh are also available in DB, as originally defined in Plummer and others, (1976). In the above example, Eh will be calculated in WATEQFP using an equilibrium relation between dissolved sulfate and dissolved hydrogen sulfide.

Data entered with <A>dd are permanently stored in the .LON file after a <S>ave (see Save function below).

Delete

With this selection, a well can be removed from the database. This change will only be permanent if the .LON file is subsequently Saved (See <S>ave function below).

Edit

When the edit screen appears, either upon completion of adding a well or selection of edit, the data for the well are displayed on the screen (see above). Missing, that is, undefined values are displayed as **********. To edit a value, enter the number next to the corresponding label. To edit the well name, as well as the owner's address, date, time, and other data related to the physical characteristics of the well, enter 41. Once selected for a particular water analysis, it is not possible to edit the units of concentration or choice of activity coefficients while running DB. If it is necessary to change these values, the user must to edit the .LON file external to DB (after <S>ave and <Q>uit). The first 4 integers of the first line of information for each water analysis of the .LON file contain the WATEQF flags IUNITS, PECALC, CORALK, and IDAVES (See Plummer and others, 1976 for definition). NOTE: Data changed with <E>dit are permanently stored in the .LON file only after a <S>ave (see Save function below).

Move

The order of the wells can be changed using this command. The number of the well to be moved and the well number after which it is to be inserted are entered, and the move is carried out.

Next Page

Only 30 wells will be displayed on the screen at once. If there are more than thirty wells in the database, <N>ext Page displays the remaining wells on the screen. This command will not be listed if there are fewer than 30 wells in the database.

Print

This option has two functions: (1) to print a report (to a file) of the data for a well, and (2) to perform a preliminary check of cation-anion balance. Printed reports can be generated for a single well or for all wells in the file. Resulting files are named OUTPUTxx, where xx is the number of the well that is printed (see Attachment F). Checking cation-anion balance produces a file with the suffix .CHECK, which lists the percent error (plus or minus) of the charge balance calculation for each well. The percent charge imbalance is calculated from the relation

$$Percent\ \ Charge\ \ Imbalance\ = \frac{\sum meq_{cations} - \sum meq_{anions}}{\sum meq_{cations} + \sum meq_{anions}}\ 100\ ,$$

where meq is milliequivalents per kilogram H_2O. Note that this calculation of charge imbalance should be used only as a rough guide, because it is based on single, assumed, predominant species for each element. WATEQFP performs a complete charge balance analysis based on the temperature, pH, and chemical speciation. The complete charge balance appears in the .OUT file which is produced in making the .PAT file.

Save

It is important to save entered data before quitting DB. Three choices are presented when using the <S>ave command: (1) save raw data (to the .LON file), (2) create a .PAT file, or (3) both (1) and (2). Choice 3, make both the .LON and .PAT file, is default. The .LON file for a system contains all the data entered for all of the wells considered. In creating the .PAT file, a special WATEQFP input file, .IN, is created and read by WATEQFP to produce the .PAT file. During this operation, Redox Ignored warnings may appear. These indicate that any iron (Fe) and manganese (Mn) results may be incorrect. Without redox information, iron and manganese can not be speciated among the possible redox states. If Fe or Mn data are given, but Eh data are not available, a default Eh of zero volts is assumed. The default value for solution density is 1.0, if not entered.

Quit

DB is terminated. If option 2 or 3 was selected in <S> ave, WATEQFP is run to create a NETPATH input file (.PAT).

Creating a NETPATH input file (.PAT)

Upon completion of the program, DB looks for an input file for the program WATEQFP (.IN) saved by DB (option 2 or 3 in <S>ave). If one is found, WATEQFP is run to create the .PAT file. If a charge balance error of more than 30 percent is encountered in running WATEQFP, or an error in pH is suspected (pH outside the interval $3 \leq pH \leq 11$), the user is prompted whether to continue making the .PAT file or to abort. Normally, if errors are encountered, the user should return to DB to correct the analytical data before proceeding to NETPATH. The .PAT file contains all the analytical and isotopic data needed to run NETPATH. Analytical data are stored in units of millimoles per kilogram H_2O in the range 0.001 to 9999.999.

Due to missing data and (or) analytical errors, chemical analyses of waters are rarely electrically charge balanced. As a first attempt to address this problem, DB allows the .PAT file to be constructed in two ways: (1) using the original unaltered analytical data which may or may not be electrically balanced (default), and (2) modifying the original analytical data to attain approximate charge balance. In this latter case the electrical imbalance of each water analysis is computed assuming the following species in solution: Ca^{2+}, Mg^{2+}, Na^{+}, K^{+}, Cl^{-}, SO_4^{2-}, HCO_3^{-}, Ba^{2+}, Sr^{2+}, Fe^{2+}, Li^{+}, F^{-}, Br^{-}, NO_3^{-}, $H_2PO_4^{-}$, Al^{3+}, NH_4^{+}, and H^{+}. This charge imbalance is then distributed proportionally on the basis of equivalents among all the above species except H^{+}. Data for boron and hydrogen sulfide are not included because their neutral species are assumed predominant. When charge balancing is selected, all original analytical data for the analysis are modified in the .PAT file, creating altered water analyses that are approximately charge balanced. The charge-balancing procedure is approximate because analytical errors in minor element concentrations are ignored and the adjustments do not take into account the full speciation/complexation of the aqueous model. There must be analytical data for at least one major cation and one major anion in the analysis for the charge balancing option to be invoked. If greater accuracy in charge balancing of analytical data is required, this should be carried out by the user prior to entering data in DB.

Caution should be exercised when selecting the charge balancing option because all the original analytical data will be modified in the .PAT file. However, none of the original data in the .LON file are altered when the charge-balance option is selected. Ideally, geochemical modeling should be conducted using perfectly analyzed, electrically balanced waters, but in practice, this is rarely the case.

Once the .PAT file is created, the WATEQFP output file is saved as [*Filename*].OUT. This file contains the full output from WATEQF (Plummer and others, 1976), including mineral saturation indices.

RUNNING NETPATH

General Notes

Wherever practical in NETPATH, <Enter> results in the most common or natural selection being chosen or the current state of the value being requested to be retained. At prompts with no single preferable option or no previous reasonable value, <Enter> usually has no effect. Therefore, when the user is presented with a series of prompts, <Enter> will quickly run through them with no changes to current values.

NETPATH also handles incorrect input without abnormal termination. Values out of range or of incorrect type are handled without improper program termination. For example, when the program expects a real (noninteger) number input, an integer will be read and converted to a real value in the program. Entering letters at a prompt for a numerical value will not cause any major problem, although the value will have to be reentered.

File Selection

Once the user has prepared a file of water analyses using DB, and created the .PAT file for these data, NETPATH may be started. In running NETPATH, a list of all the available .PAT files of data (listed in NETPATH.FIL) is first displayed, and the user is instructed to select one. Next, NETPATH displays a list of model files (stored in MODEL.FIL) previously saved, if present, under the selected well file. If no model file is selected (or present), NETPATH requests whether the problem involves mixing of initial waters and then requests selection of initial and final wells from the .PAT file. If a previously saved model file is selected, lists of constraints or lists of phases in the model are displayed along with the wells to be modeled. The user has the option of selecting the model or selecting phases and constraints individually later. If an existing model file is selected, the user has the option of accepting the model, deleting it from the list, or returning to the model list for another selection. If accepted, the phases and constraints read in may later be modified by choosing the appropriate options from the main screen of NETPATH.

Main Screen Functions

Once the desired files or wells have been selected, the main screen is displayed. This screen contains many of the important facts about the current model under consideration. When initially setting up a model, the screen is blank under the headings Constraints and Phases (see Add below). The currently selected wells, the constraints and phases being used, and some parameters are displayed. By responding to a few prompts, any part of the model can be edited. Warnings may be displayed if inappropriate choices are made. This is a sample main screen:

```
Initial Well:Recharge #3
Final Well  :Mysse
==============================================================================
     Constraints: 10       |         Phases: 10          |      Parameters
---------------------------|----------------------------|----------------------
Carbon       Sulfur        |+DOLOMITE CALCITE +GYPSUM   | Mixing: No
Calcium      Magnesium     |+"CH2O"   GOETHITE-PYRITE   | Evaporation: No
Sodium       Potassium     | EXCHANGE+NaCl      SYLVITE | Rayleigh Calcs: Yes
Chloride     Ferrous       | CO2 GAS                    | Exchange: Ca/Na
Redox        Sulfur-34     |                            | Init C-14      52.33
                           |                            | (Mass Balance)
==============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit
```

From the main screen, the user can branch to <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit. A description of each of these options follows.

Add

This selection is used to add constraints or phases. Constraints and phases were discussed earlier in the section IMPORTANT CONCEPTS IN NETPATH. All available element, redox and isotopic constraints are listed for selection.

A selection of commonly used phases is stored in the file NETPATH.DAT. This file is read by NETPATH and any mineral in the file can be selected. As the prompt states, typing L at the prompt will return a list of the phases read from NETPATH.DAT, their corresponding numbers, and their default precipitation-dissolution limitations. Typing the number of a phase at the <A>dd phase prompt will select a previously stored phase to be considered as part of the model. If the desired phase is not included in NETPATH.DAT, the desired phase and its stoichiometric coefficients can be entered and the resulting phase saved into NETPATH.DAT for later use.

After a phase is selected or defined, the user is presented with some questions about the phase. Usually, the default, selected by <Enter>, is all that is needed. Each phase must be marked in one of four ways: (1) dissolution only, (2) precipitation only, (3) dissolution or precipitation allowed, and (4) isotope exchange. Options 1-2 limit the total possible number of models. Sometimes it is not known whether the phase is a reactant or product, in which case option 3 is selected. Option 4, isotope exchange, allows the user to specify an amount (mmol/kg H_2O) of a phase that dissolves and precipitates in an isotopic equilibrium exchange. If isotopic exchange is selected for a phase, the phase will also be allowed to dissolve or precipitate a net mass transfer along the flow path. Another option allows the user to specify whether every model must contain that phase (termed forcing). Below is an example of the main screen showing kaolinite forced to be included in every model and allowing dissolution only (+) for NaCl and gypsum, and precipitation only, (-), for kaolinite and Ca-montmorillonite:

```
Initial Well:Sierra Nevada (Ephemeral Spr.)
Final Well  :Sierra Nevada (Perennial Spr.)
============================================================================
     Constraints:  8       |         Phases:  9          |      Parameters
-------------------------|-----------Forced-----------|--------------------
Carbon        Sulfur       |-KAOLINIT                    | Mixing: No
Calcium       Aluminum     |----------Unforced----------| Evaporation: No
Sodium        Chloride     |+NaCl    +GYPSUM   BIOTITE  | Rayleigh Calcs: No
Silica        Magnesium    | PLAGAN38-Ca-MONT  CO2 GAS  |
                           | CALCITE  SiO2              |
============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit
```

Delete

The < D > elete option allows phases or constraints to be removed from consideration in the model. More than one constraint or phase may be deleted before returning to the main screen. Within the Delete option the only way to delete all the phases or constraints is one at a time from the list displayed.

Edit

This selection is used to change a wide range of conditions and parameters in the models. Initiation of < E > dit prints an Edit menu on the screen, for example:

```
General
  1) Well file              : MADISON
  2) Entire model
  3) Phases
Wells
  4) Mixing                 : No
  5) Initial well           : Recharge #3
  6) Final well             : Mysse
Parameters
  7) Evaporation/Dilution   : No
  8) Ion exchange           : Ca/Na
  9) Redox state of DOC
Isotope calculations
 10) Rayleigh calculations  : Yes
 11) Isotopic data
 12) Model for initial C14  : Mass Balance
 13) Carbon fract. Factors  : Mook
-------------------
Edit which? (<Enter> when done)
```

By selecting the appropriate number from the Edit menu it is possible to (1) change the well file to another set of wells (if other .PAT files exist), (2) change the model, if models have been previously saved under the current well file, (3) change or edit phases, (4) switch between mixing and nonmixing models, (5)-(6) select different initial and final waters from the well file, (7) switch on or off the evaporation/dilution option, (8) select various options affecting cation exchange, (9) redefine the redox state of DOC in initial and final waters (for the current model), (10) switch on or off the Rayleigh calculations and, if the Rayleigh calculations are invoked, (11) edit isotopic data on phases, isotopic values of dissolved methane and DOC, and isotopic fractionation factors, (12) select a method for defining the initial ^{14}C content, A_0, for radiocarbon dating, and (13) select either the Mook (1980) or Deines and others (1974) sets of fractionation factors for the inorganic carbonate system. The above numbers of options in the Edit menu can change depending on selection of phases (EXCHANGE and CO2-CH4), and whether mixing or Rayleigh calculations are called for. If the phase CO2-CH4 is added, an additional parameter is displayed for definition of the fraction of CO_2 gas in the mixture. NETPATH returns to the edit menu after completion of each task. Some of the general features of <E>dit are described below.

Editing well files

This option allows the user to select another .PAT file from the well file (listed in NETPATH.FIL). If no other .PAT file exists in the directory, only the current .PAT file can be selected. When changing .PAT files, a previously defined model is not retained.

Editing model files

This option allows the user to return to the list of models (in NETPATH.FIL) previously saved under the current well file and select another model file. Once selected, the set of constraints and phases previously defined for the model are displayed and the user has the option of deleting the model from the list of models, selecting another model, or accepting the model.

Editing phases

After this option is chosen, a list of the current phases in the model is displayed. A phase can be selected from the list by number and then edited. The entire phase can be replaced by either a previously entered phase or a user-entered phase, or simply kept. Selecting "Phases" from the Edit menu allows the user to modify any phase in the model, changing its name, stoichiometry, and dissolution, precipitation and forcing attributes. In defining new names of phases, the only limitations are that "CO2" should not be the first three letters of the phase name and the phase should not be called "EXCHANGE". These keywords initiate phase-specific routines that are not appropriate for most phases. To include additional phases in the model, use <A>dd.

Once a name has been entered, constraints can be added one at a time, along with their coefficients in the phase. Once this process is over, the selected constraints and their coefficients are displayed, and they may be edited, deleted, or more constraints can be added. <Enter> ends this editing. Once dissolution/precipitation and forced/unforced information has been entered, the phase may be stored in NETPATH.DAT. If the name is the same as a phase already in the file, it can be saved under a new name or it can replace the current phase in NETPATH.DAT. The initial designation of dissolution/precipitation behavior is stored as a default value for the phase and recalled each time the phase is added to a model. This default designation can be changed by editing the phase in the model and (or) overwriting the phase in the NETPATH.DAT file with a new default condition. If the program is terminated in any way other than <Q>uit, these additions to NETPATH.DAT will not be saved.

Editing mixing

With this option, the user can select whether mixing is to be considered. If mixing is invoked, the list of wells from the well file (NETPATH.FIL) is displayed for selection of the second initial water. If the mixing option is turned off, the second initial well is eliminated.

Editing wells (selecting wells within the well file)

Options 5 and 6 (and 7, if mixing) allow the user to select new initial and final wells from the [*Filename*].PAT file. Editing wells provides a means to apply a previously defined model to a new set of initial and final water analyses. <Enter> at any of the prompts keeps the current value, unless no value has been previously entered.

Editing parameters

Parameters can be used to include the possibility of evaporation or dilution (with pure water) during the evolution of the initial water(s) to the final water, define various ways cation exchange can be calculated among Ca^{2+}, Mg^{2+}, and Na^{+} in the model, define the fraction of CO_2 gas in the CO2-CH4 gas mixture, and adjust the redox state of the DOC for each well within the model. Changes to the redox state of DOC in NETPATH do not effect the original data stored in either the .LON or .PAT files. Current values of all parameters are saved for a model using the <S>ave option from the main screen. The phases EXCHANGE and CO2-CH4 must be present in the model before they can be edited through the Edit menu.

Editing isotope calculations

A toggle to allow Rayleigh calculations is accessed through the Edit menu. Once Rayleigh calculations are allowed, isotopic data may be edited. The data entered into this part of the program are used both for isotope mass balance and Rayleigh calculations. Even if the problem is one of the strictly valid isotope mass balance cases discussed above (see ISOTOPIC CALCULATIONS), it is still necessary to enable the Rayleigh calculations to <E>dit the isotopic data. If Rayleigh calculations are selected, and phases containing carbon, sulfur, or strontium are included in the model, three more options appear on the Edit menu, enabling the user to (1) edit isotopic compositions and fractionation factors, (2) select various means of defining the initial ^{14}C activity, A_o, in the recharge water, and (3) select one of two internal sets of carbonate-system fractionation factors (Mook, 1980; Deines and others, 1974).

The option <E>dit, Isotopic data allows values for the $\delta^{13}C$, ^{14}C (pmc), $\delta^{34}S$, and $^{87}Sr/^{86}Sr$ to be defined for all carbon, sulfur and Sr-bearing phases in the model. Control passes to a second screen under this option to allow editing of the carbon isotopic composition of dissolved methane and DOC, if either are present in any of the selected wells. Finally a third screen appears to allow editing of fractionation factors. The next choice from the Edit menu allows definition of the initial ^{14}C content (A_0 in percent modern carbon), or calculation of A_0 from some of the various models available in the literature. The final choice from the Edit menu allows selection of either the Mook (1980) or Deines and others (1974) sets of fractionation factors of inorganic carbon. The editing of isotopic data is discussed in more detail below.

Editing the isotopic composition of phases. -- The first screen under <E>dit, Isotopic data deals with the isotopic composition of phases selected on the main screen. If there are any phases containing carbon, sulfur, or strontium, the first screen displayed under the <E>dit, Isotopic data option contains the isotopic compositions of $\delta^{13}C$, ^{14}C (pmc), $\delta^{34}S$, and $^{87}Sr/^{86}Sr$ in those phases. An example of this screen is as follows:

```
                         Isotopic Compositions

Number  Phase      Carbon-13   C-14 (% mod)    Sulfur-34   Strontium-87
------------------------------------------------------------------------
    1:  DOLOMITE      4.0000       0.0000
    2:  CALCITE       0.0000       0.0000
    3:  GYPSUM                                   15.5000
    4:  "CH2O"      -25.0000       0.0000
    5:  PYRITE                                  -22.0900
    6:  CO2 GAS     -25.0000     100.0000

Enter number of phase to edit. <Enter> when done.
```

In this example there were no Sr-bearing phases in the model.

The isotopic compositions of the phases have either been entered by the user, retrieved from a saved phase file, retrieved from NETPATH.DAT, or they are undefined. If "CO2-CH4" is a selected phase, $\delta^{13}C$ and ^{14}C values will be displayed separately for the two gases in the gas mixture. The values will be combined according to the ratio determined by the fraction of CO_2 in the phase (see previous discussion of the CO_2-CH_4 phase). The listed isotopic compositions will be used in mass-balance calculations if $\delta^{13}C$, ^{14}C, $\delta^{34}S$, or $^{87}Sr/^{86}Sr$ are included as constraints. These isotopic compositions apply to both dissolution and precipitation (outgassing) processes when the isotope is included as a constraint. Alternatively, if Rayleigh calculations are made, these isotopic values are used for incoming sources only. Dissolution is considered a non-fractionating process. The $\delta^{13}C$, ^{14}C, $\delta^{34}S$, and $^{87}Sr/^{86}Sr$ isotopic compositions of all outgoing phases containing these elements are determined according to defined fractionation factors. The isotopic compositions have the following units: $\delta^{13}C$ in per mil relative to PDB, ^{14}C in percent modern carbon (pmc) of the NBS oxalic acid standard, $\delta^{34}S$ in per mil relative to CDT, and Strontium-87 as the isotope ratio, $^{87}Sr/^{86}Sr$.

Editing the isotopic composition of dissolved methane and dissolved organic carbon. -- If the concentrations of dissolved methane and (or) DOC were entered in DB, they are stored in the .PAT file for use in NETPATH where they are summed with the dissolved inorganic carbon in defining the total dissolved carbon in solution. For isotopic calculations of $\delta^{13}C$, it is also necessary to know the $\delta^{13}C$ composition of the dissolved methane and DOC. Because these data are not routinely measured, there is no provision to store them through DB and into the .PAT file. Instead NETPATH accepts data for the $\delta^{13}C$ composition of dissolved methane and DOC for a particular model. If data are given in DB for the concentrations of dissolved methane and (or) DOC, and carbon is selected as a constraint in the NETPATH model, a second screen appears under <E>dit, Isotopic data for entering (and adjusting) the $\delta^{13}C$ composition of dissolved methane and DOC. An example of this screen follows:

```
                      Isotopic compositions of Carbon in solution
                                          Carbon-13  C14 %mod   Carbon-13  C14 %mod
#  Well Name                                of CH4    of CH4      of DOC    of DOC
---------------------------------------------------------------------------------
1: Nashville #4                                                  -25.000     0.000
2: FAA Hilliard                                                  -25.000     0.000
Enter number of well to change, <Enter> when done.
```

In the above example, the .PAT file contained no data for the concentration of dissolved methane. The .PAT file does contain analytical data for DOC and here its $\delta^{13}C$ is defined to be -25 per mil and its ^{14}C content is defined to be 0 percent modern.

Editing fractionation factors. --Editing of fractionation factors is accessed through the third screen which appears under the < E > dit Isotope Data option of the Edit menu. Additive fractionation factors, ϵ, may be specified for any precipitating (or outgassing) phase containing carbon, sulfur, or strontium.

Although fractionating processes may be included for the $^{87}Sr/^{86}Sr$ ratio, isotope fractionation is not expected for strontium owing to the relatively high mass. The strontium isotope ratio is included with carbon and sulfur isotopes because all are treated using isotope evolution equations (Wigley and others, 1978, 1979) when Rayleigh calculations are invoked. The default additive fractionation factor for $^{87}Sr/^{86}Sr$ of 0.0 should be utilized in this case. If sufficient data are available in the DB database, NETPATH computes default additive fractionation factors for $\delta^{13}C$, $\delta^{14}C$, and $\delta^{34}S$ relative to the average isotopic composition of that element in solution. The calculated per mil equilibrium additive fractionation factor for ^{14}C species is taken as two-fold the ^{13}C fractionation (Craig, 1954). Alternatively, the ^{14}C fractionation factors can be defined under < E > dit Isotope Data.

The additive fractionation factors are calculated for the conditions at the final well (default X = 1), but may be specified at the initial well conditions (X = 0), or at any value in the interval $0 \leq X \leq 1$ where X is the fraction of the flow path between initial and final well. In NETPATH, fractionation factors calculated based on well data are shown with an asterisk. In mixing cases the initial fractionation factor is defined as the mean of the fractionation factors for the two initial waters. An example of the second isotope screen displayed for definition of fractionation factors is as follows:

```
              Additive Fractionation Factors (in per mil)
                          Relative to solution
Number  Phase      Carbon-13     Carbon-14    Sulfur-34   Strontium-87
-----------------------------------------------------------------------

   1:  CALCITE       3.8708*       7.7416*
   2:  PYRITE                                  -37.8958*
   3:  CO2 GAS      -2.9695*      -5.9390*

* = based on computed value at 1.00 fraction between init and final waters

Enter number of phase to edit, '0' for fraction along path, <Enter> when done.
```

Run

Selection of <R>un executes a modified version of BALANCE (Parkhurst and others, 1982) which has been named NEWBAL (Parkhurst, unpublished data on file in the Branch of Regional Research, Central Region, U.S. Geological Survey). NETPATH first displays the data to be input to NEWBAL, with prompts to abort should unexpected data be recognized. The following abbreviations are used in the display of the input data to the mass balance:

Abr.	Long Name	Abr.	Long Name	Abr.	Long Name	Abr.	Long Name
18	Oxygen-18	D	Deuterium	LI	Lithium	RS	Redox
AK	Alkalinity	F	Fluoride	MG	Magnesium	S	Sulfur
AL	Aluminum	FE	Iron	MN	Manganese	SH	34SH2S
B	Boron	HS	Sulfide	NA	Sodium	SI	Silica
BA	Barium	I1	Carbon-13	NH	Ammonium	SO	Sulfate
BR	Bromide	I2	C-14 (% mod)	NO	Nitrate	SR	Strontium
C	Carbon	I3	Sulfur-34	O2	Diss. O2	SS	34SSO4
CA	Calcium	I4	Sr-87/Sr-86	OC	Organic C	TE	Temp. (C)
CL	Chloride	K	Potassium	P	Phosphate	TR	Tritium

This information can be used to check that the appropriate compositions of phases are being used. Also listed as input to NEWBAL are the compositions of the initial and final waters in mmol/kg H_2O.

All combinations of phases are checked to see if mass transfers can be found that satisfy the chosen constraints. The total number of models found, if any, is then displayed. If many more phases are included than constraints, the total number of models can be very large. Relatively long computation times may be required to find all possible models. While running NEWBAL, the screen displays the total number of models to be tested, the cumulative total number of models tested (update in increments of 100 models), and the updated total number of models found. As discussed in IMPORTANT CONCEPTS IN NETPATH, the total number of models can be reduced by deleting unnecessary phases, editing phases by marking them for dissolution or precipitation only, and including appropriate forcing conditions for phases.

Alternatively, if no models are found, the program will determine if models can be found by ignoring precipitation/dissolution constraints. The output indicates which constraints could be ignored to find models. This procedure is sometimes useful in guiding the modeling exercise. If only one model is found, it will be displayed. If more than one model is found, the user is prompted whether to display the models all at once, one at a time, or not at all. At any prompt in this section, the user may return to the main screen.

Isotope calculations during run

When the Rayleigh calculations are enabled (through the Edit menu), isotopic values are calculated for each of the four isotopes $\delta^{13}C$, $\delta^{14}C$, $\delta^{34}S$, and $^{87}Sr/^{86}Sr$. An error message is displayed if there is some difficulty, such as missing data that do not allow the calculation to be performed. It may be necessary to return to DB to add appropriate isotope data to the water analysis, save a new .LON file, and make a new .PAT file. Alternatively, the error message could result from failure to define appropriate isotopic data required in making the Rayleigh calculations. In this case it is necessary to return to <E>dit, Isotopic data to add the missing data.

After a model is displayed, more details about the isotopic evolution model can be displayed. The data input into the Rayleigh calculations can be displayed by pressing <S>. This operation also displays the initial Carbon-13 and initial Carbon-14 values for total dissolved carbon in the initial solution, and the initial Sulfur-34 per mil value for total dissolved sulfur in the initial solution. An example display is as follows:

```
Data used for Carbon-13
Initial Value:    -6.9900
 2 dissolving:
    DOLOMITE  Delta element:   7.06000  Isotopic composition:    4.0000
    "CH2O"    Delta element:   0.87067  Isotopic composition:  -25.0000
 2 precipitating:
    CALCITE   Delta element:   5.31723  Fractionation factor:    2.3815
    CO2 GAS   Delta element:   0.04345  Fractionation factor:   -5.9230
Data used for C-14 (permil)
Initial Value:    52.3256
 2 dissolving:
    DOLOMITE  Delta element:   7.06000  Isotopic composition:-1000.0000
    "CH2O"    Delta element:   0.87067  Isotopic composition:-1000.0000
 2 precipitating:
    CALCITE   Delta element:   5.31723  Fractionation factor:    4.7631
    CO2 GAS   Delta element:   0.04345  Fractionation factor:  -11.8460
Data used for Sulfur-34
Initial Value:    9.7300
 1 dissolving:
    ANHYDRIT  Delta element:  20.14723  Isotopic composition:   15.5000
 1 precipitating:
    PYRITE    Delta element:   0.18823  Fractionation factor:  -37.8958
Data used for Strontium-87
      Insufficient data
No more models, <S> to show Rayleigh data, <C> to run all C-14 models,
<Enter> to continue
```

To run all of the ^{14}C models and calculate the corresponding ages, <C> can be entered. An example of this screen is as follows:

```
Model                        A0     Computed  Observed     Age
                                   (no decay)
------------------------------------------------------------
Original Data              33.05      7.75        .80     18772
Mass Balance               52.33     12.27        .80     22570
Vogel                      85.00     19.93        .80     26580
Tamers                     53.46     12.54        .80     22748
Ingerson and Pearson       52.33     12.27        .80     22570
Mook                       53.80     12.62        .80     22800
Fontes and Garnier         46.52     10.91        .80     21598
Eichinger                  47.57     11.15        .80     21782
User-defined              100.00     23.45        .80     27924
```

Save

NETPATH allows the user to save two types of files, (1) the model file and (2) the results of the mass balance calculation. The saved model file contains all wells, data, parameters, selections, and attributes needed to repeat the identical calculation, and may be retrieved interactively by selecting <E>dit, Well file, or upon initiation of a new NETPATH session. The name of the model file is specified by the user after selecting <S>ave. It is important to save the model file before executing the <Q>uit command, if it is anticipated that the model could be used in subsequent calculations.

The results of the mass balance and isotopic calculations can also be saved to a separate file, but can always be reproduced if the model file has been saved. In some cases NETPATH reports models that exist only if certain mineral dissolution or precipitation constraints are ignored. Although reaction information is displayed on the screen when models are found only by ignoring dissolution/precipitation constraints, the information is not saved to a results file. In this case the results file would indicate that no models were found. In saving a model or result, if a previous file already exists for the model or result, the filename is returned to the screen and the user has the option to overwrite the existing file, or specify a new filename.

Quit

Once all modeling is completed and all desired files saved, the user selects <Q>uit to exit the program. The changes made to NETPATH.DAT are written at this point, so it is important that this option be used to exit the program.

EXAMPLES AND TEST PROBLEMS

This section treats 7 examples that serve the dual functions of demonstrating specific applications of NETPATH and testing code performance on other computers. The diskette in the back pocket of this report contains model files that reproduce the results of the examples that appear below. The file names are listed in the file MODEL.FIL on the diskette, and identifed in the text below.

Example 1: Silicate Weathering in the Sierra Nevadas

In their classic paper "Origin of the Chemical Compositions of some Springs and Lakes", Garrels and Mackenzie (1967) performed a chemical mass balance on two spring compositions from the Sierra Nevada reported by Feth and others (1964). The two spring compositions are given below:

[*Analyses of mean values in millimoles per liter from Garrels and Mackenzie (1967)*]

	pH	*SiO_2*	*Ca^{2+}*	*Mg^{2+}*	*Na^+*	*K^+*	*HCO_3^-*	*SO_4^{2-}*	*Cl^-*
Ephemeral Spr.	6.2	0.273	0.078	0.029	0.134	0.028	0.328	0.010	0.014
Perennial Spr.	6.8	.410	.260	.071	.259	.040	.895	.025	.030

Garrels and Mackenzie performed a "reverse" mass balance reconstituting the original granitic rock from the differences in water composition. The mass-balance approach used here and "reverse" mass-balance approach of Garrels and Mackenzie (1967) are identical in principle. From the "reverse" mass-balance approach the result is (Garrels and Mackenzie (1967):

[*Mass transfer in millimoles per kilogram H_2O*]

Phase	*Composition*	*Mass Transfer*
"Halite"	$NaCl$	0.016
"Gypsum"	$CaSO_4$	.015
Kaolinite	$Al_2Si_2O_5(OH)_4$	-.033
Ca-Montmorillonite	$Ca_{0.17}Al_{2.33}Si_{3.67}O_{10}(OH)_2$	-.081
CO_2 gas	CO_2	.427
Calcite	$CaCO_3$	.115
Silica	SiO_2	.0
Biotite	$KMg_3AlSi_3O_{10}(OH)_2$	.014
Plagioclase	$Na_{0.62}Ca_{0.38}Al_{1.38}Si_{2.62}O_8$	.175

This problem repeats the calculation of Garrels and Mackenzie (1967) in NETPATH using the Ephemeral spring as the initial water and the Perennial spring as the final water. Garrels and Mackenzie (1967) adjusted the difference in bicarbonate concentrations to attain charge balance. In this problem the mass balance is solved using the original (charge imbalanced) waters and then repeat the final calculation selecting the charge balancing option of DB in constructing the .PAT file. A temperature of 25 °C was assumed for both waters (actual temperature not given). The phase compositions chosen by Garrels and Mackenzie are given above.

WATEQFP was used to calculate the total inorganic carbon. Aluminum was taken to be zero (actually just enter <Return> in DB if a value is unknown; the default for unknown concentrations is zero in NETPATH). The following constraints and phases were included in NETPATH:

[See file **NP-1a.dat** on the diskette]

```
Initial Well:Sierra Nevada (Ephemeral Spr.)
Final Well  :Sierra Nevada (Perennial Spr.)
===============================================================================
     Constraints: 9       |         Phases: 9         |       Parameters
------------------------- |---------------------------|----------------------
Carbon      Sulfur        |+NaCl    +GYPSUM   KAOLINIT | Mixing: No
Calcium     Magnesium     | Ca-MONT  CO2 GAS  CALCITE  | Evaporation: No
Sodium      Chloride      | SiO2     BIOTITE +PLAGAN38 | Rayleigh Calcs: No
Silica      Aluminum      |                           |
Potassium                 |                           |
===============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit
```

The data to be entered into NEWBAL are displayed by NETPATH (concentrations in the initial and final waters are in millimoles per kilogram H_2O):

```
Initial Well : Sierra Nevada (Ephemeral Spr.)
Final Well   : Sierra Nevada (Perennial Spr.)
         Final     Initial
C       1.1990       .7810
S        .0250       .0100
CA       .2600       .0780
MG       .0710       .0290
NA       .2590       .1340
CL       .0300       .0140
SI       .4100       .2730
AL       .0000       .0000
K        .0400       .0280
Hit <Q> to quit, or <Enter> to continue

NaCl    NA  1.0000CL  1.0000
GYPSUM  CA  1.0000S   1.0000RS  6.0000I3 22.0000
KAOLINITAL  2.0000SI  2.0000
Ca-MONT CA  0.1670AL  2.3300SI  3.6700
CO2 GAS C   1.0000RS  4.0000I1-25.0000I2100.0000
CALCITE CA  1.0000C   1.0000RS  4.0000I1   .0000I2   .0000
SiO2    SI  1.0000
BIOTITE AL  1.0000MG  3.0000K   1.0000SI  3.0000
PLAGAN38CA  0.3800NA   .6200AL  1.3800SI  2.6200
Hit <Q> to quit, or <Enter> to continue
```

Notice that in the above listing of phase data some information is given for RS, I1, I2, and I3 that was not used in this problem. Possible redox reactions were not considered (though pyrite oxidation and precipitation of FeOOH; and perhaps aerobic oxidation of organic matter might be included). The isotopes were not considered in this problem either (notice on the main screen that Rayleigh Calcs: "No" is given). Isotopes were not included in the mass-balance calculations because they were not chosen as constraints. Nevertheless default isotopic (and RS) values are stored for each appropriate mineral in the file NETPATH.DAT. The parameters I1 and I2 are $\delta^{13}C$ (per mil) and ^{14}C (pmc), and I3 is $\delta^{34}S$ (per mil).

Returning to our problem there are 9 constraints and 9 phases. One combination of phases exists but does not satisfy the constratints:

```
          1 models to be tested
          1 models were tested.
          0 models were found which satisfied the constraints.
No more models, hit <Enter> to continue
```

No models were found. When this occurs it is necessary to re-examine the chosen phases, and how they can satisfy the chemical variation in the chosen constraints. In this case the problem is fairly simple to see. Notice that potassium and magnesium have only one mineral source, biotite. If K and Mg are only derived from biotite then the change in concentrations of Mg and K in the waters must be in proportion to their stoichiometry in biotite, 3:1. Examination of the water composition shows that this is not exactly so. Therefore, NEWBAL finds no models. There are several ways around this problem which must be guided by geological, hydrological, and geochemical intuition. Some of the possibilities are (1) there may be more K and (or) Mg phases (such as K-spar), (2) the two waters may not be truly evolutionary, or (3) there may be small analytical error in K or Mg. If K-spar is added to the phases, the models precipitate K-spar rather than dissolve it (very unlikely for these dilute waters). Inspection of the analysis of Garrels and Mackenzie (1967) shows that these authors actually ignored the potassium in their calculation, basing the amount of biotite reacted on the change in magnesium concentration. The same logic needs to be applied in NETPATH. This is done by deleting the K constraint. The revised main screen is now:

[See file **NP-1b.dat** on the diskette]

```
Initial Well:Sierra Nevada (Ephemeral Spr.)
Final Well  :Sierra Nevada (Perennial Spr.)
==============================================================================
     Constraints:  8        |         Phases:  9         |      Parameters
--------------------------- |--------------------------- |----------------------
Carbon       Sulfur         |+NaCl    +GYPSUM   KAOLINIT |Mixing: No
Calcium      Magnesium      | Ca-MONT  CO2 GAS  CALCITE  |Evaporation: No
Sodium       Chloride       | SiO2     BIOTITE +PLAGAN38 |Rayleigh Calcs: No
Silica       Aluminum       |                            |
==============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit
```

The revised input to NEWBAL is as follows:

```
Initial Well : Sierra Nevada (Ephemeral Spr.)
Final Well   : Sierra Nevada (Perennial Spr.)
          Final     Initial
C        1.1990       .7810
S         .0250       .0100
CA        .2600       .0780
MG        .0710       .0290
NA        .2590       .1340
CL        .0300       .0140
SI        .4100       .2730
AL        .0000       .0000
Hit <Q> to quit, or <Enter> to continue
```

```
NaCl    NA  1.0000CL  1.0000
GYPSUM  CA  1.0000S   1.0000RS  6.0000I3 22.0000
KAOLINITAL  2.0000SI  2.0000
Ca-MONT CA   .1670AL  2.3300SI  3.6700
CO2 GAS C   1.0000RS  4.0000I1-25.0000I2100.0000
CALCITE CA  1.0000C   1.0000RS  4.0000I1   .0000I2   .0000
SiO2    SI  1.0000
BIOTITE AL  1.0000MG  3.0000K   1.0000SI  3.0000
PLAGAN38CA   .3800NA   .6200AL  1.3800SI  2.6200
Hit <Q> to quit, or <Enter> to continue
```

Nine models were tested and five found that satisfied the chemical constraints:

```
           9 models to be tested
           9 models were tested.
           5 models were found which satisfied the constraints.
Display models: <A>ll at once, [ENTER] for each, or <N>one.

           MODEL   1
 NaCl      +           .01600
 GYPSUM    +           .01500
 KAOLINIT             -.03354
 Ca-MONT              -.08134
 CO2 GAS               .30422
 CALCITE               .11378
 BIOTITE               .01400
 PLAGAN38  +           .17581
<Enter> to continue, any other key to quit

           MODEL   2
 NaCl      +           .01600
 GYPSUM    +           .01500
 KAOLINIT             -.82726
 Ca-MONT               .59996
 CO2 GAS               .41800
 SiO2                 -.91295
 BIOTITE               .01400
 PLAGAN38  +           .17581
<Enter> to continue, any other key to quit

           MODEL   3
 NaCl      +           .01600
 GYPSUM    +           .01500
 KAOLINIT             2.08873
 Ca-MONT             -1.90303
 CALCITE               .41800
 SiO2                 2.44106
 BIOTITE               .01400
 PLAGAN38  +           .17581
<Enter> to continue, any other key to quit

           MODEL   4
 NaCl      +           .01600
 GYPSUM    +           .01500
 KAOLINIT             -.12831
 CO2 GAS               .31781
 CALCITE               .10019
 SiO2                 -.10900
 BIOTITE               .01400
 PLAGAN38  +           .17581
<Enter> to continue, any other key to quit
```

```
             MODEL   5
 NaCl       +           .01600
 GYPSUM     +           .01500
 Ca-MONT               -.11013
 CO2 GAS                .29941
 CALCITE                .11859
 SiO2                   .03858
 BIOTITE                .01400
 PLAGAN38   +           .17581
No more models, hit <Enter> to continue
```

All five models dissolve biotite and plagioclase, as expected (because no other sources of Mg and Na were included). Kaolinite and Ca-montmorillonite are expected to be products, rather than reactants, in this weathering environment, therefore models 2 and 3 are probably not valid (though this opens the question of the thermodynamics of clay minerals and their dissolution behavior). Because kaolinite is expected to be a weathering product in this system, model 5 is probably unrealistic. This narrows models among the selected phases down to 1 and 4. Is dissolved silica conserved among the aluminosilicate phases or is some form of silica a product in the system? Returning to the WATEQFP output generated by DB it is seen that both waters are oversaturated with respect to quartz, and even more so if the waters are colder than the assumed 25 °C. The final water (Perennial spring) is oversaturated with chalcedony (SI = 0.14 and 0.32 at 25 and 10 °C, respectively). The initial water (Ephemeral spring) is very close to saturation with chalcedony (SI = -0.04 and 0.14 at 25 and 10 °C, respectively). The conclusion from WATEQFP is that a common form of silica could precipitate from the Sierra Nevada waters. Thus, either models 1 or 4 are most likely for the chosen phases. Petrographic evidence for the abundance of Ca-montmorillonite would help resolve this question.

We could have used some of the above reasoning in the initial selection of phases to narrow the possibilities. For example it could required that kaolinite be included in every model, and both kaolinite and Ca-montmorillonite be products in the system. This can be accomplished in NETPATH Using <E>dit, Phase. The revised screen shows that kaolinite is "Forced" to be included in every model and the minus signs in front of kaolinite and Ca-montmorillonite mean that only models that precipitate these phases will be considered. The plus signs for NaCl, gypsum, and plagioclase mean that only models dissolving these phases will be displayed.

[See file **NP-1c.dat** on the diskette]

```
Initial Well:Sierra Nevada (Ephemeral Spr.)
Final Well  :Sierra Nevada (Perennial Spr.)
==============================================================================
     Constraints:  8          |          Phases:  9         |      Parameters
---------------------------   |------------Forced-----------|-----------------------
Carbon        Sulfur          |-KAOLINIT                    |Mixing: No
Calcium       Magnesium       |----------Unforced-----------|Evaporation: No
Sodium        Chloride        |+NaCl    +GYPSUM   -Ca-MONT  |Rayleigh Calcs: No
Silica        Aluminum        | CO2 GAS  CALCITE   SiO2     |
                              | BIOTITE +PLAGAN38           |
==============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit
```

The calculations below show that eight models were tested of which only two meet the requirements.

```
Initial Well : Sierra Nevada (Ephemeral Spr.)
Final Well   : Sierra Nevada (Perennial Spr.)
         Final     Initial
C       1.1990       .7810
S        .0250       .0100
CA       .2600       .0780
MG       .0710       .0290
NA       .2590       .1340
CL       .0300       .0140
SI       .4100       .2730
AL       .0000       .0000
Hit <Q> to quit, or <Enter> to continue

KAOLINITAL  2.0000SI  2.0000
NaCl    NA  1.0000CL  1.0000
GYPSUM  CA  1.0000S   1.0000RS  6.0000I3 22.0000
Ca-MONT CA   .1670AL  2.3300SI  3.6700
CO2 GAS C   1.0000RS  4.0000I1-25.0000I210 .0000
CALCITE CA  1.0000C   1.0000RS  4.0000I1   .0000I2   .0000
SiO2    SI  1.0000
BIOTITE AL  1.0000MG  3.0000K   1.0000SI  3.0000
PLAGAN38CA   .3800NA   .6200AL  1.3800SI  2.6200
Hit <Q> to quit, or <Enter> to continue

           8 models to be tested
           8 models were tested.
           2 models were found which satisfied the constraints.
Display models: <A>ll at once, [ENTER] for each, or <N>one.

           MODEL   1
 KAOLINIT  - F         -.03354
 NaCl      +            .01600
 GYPSUM    +            .01500
 Ca-MONT   -           -.08134
 CO2 GAS                .30422
 CALCITE                .11378
 BIOTITE                .01400
 PLAGAN38  +            .17581
<Enter> to continue, any other key to quit

           MODEL   2
 KAOLINIT  - F         -.12831
 NaCl      +            .01600
 GYPSUM    +            .01500
 CO2 GAS                .31781
 CALCITE                .10019
 SiO2                  -.10900
 BIOTITE                .01400
 PLAGAN38  +            .17581
No more models, hit <Enter> to continue
```

The results of the silica-conserved model (model 1) are again nearly identical to that of Garrels and Mackenzie (1967). The only significant difference is in the CO_2-gas mass transfer. The difference follows from the fact that Garrels and Mackenzie (1967) made a mass balance on HCO_3^-, and NETPATH makes the mass balance on TDC. For the same amount of bicarbonate, the Ephemeral Spring would have more total dissolved carbon than the Perennial Spring because of the lower pH and greater concentration of $CO_{2(aq)}$. The results of NETPATH take this difference into account by considering the total dissolved carbon in the system. The CO_2-gas mass transfer cannot be correctly calculated from the bicarbonate balance alone. In addition, NETPATH found another model (model 2) that produces Si_2 rather than Ca-montmorillonite. In reality the waters may evolve as a linear combination of models 1 and 2.

The validity of the final model(s) depends in part on our selection of phases. For every change in the anorthite mole fraction of the plagioclase, there is another model. How well is the plagioclase composition known? How much actual variation is there in the plagioclase composition? What are the actual clay compositions? The more that is known about the compositions of the phases in the system, the more that can be learned about the mass transfer.

Example 2: Evaporation of Dilute Inflow in Origin of Great Salt Lake, Utah

In this admittedly simplified problem, Great Salt Lake (GSL), Utah is modeled as a closed-basin lake resulting from evaporative concentration of dilute inflow from rivers with accompanying mineral precipitation. This problem uses NETPATH to determine the evaporation factor necessary to produce GSL water and mass transfer of a given set of phases. Water analyses have been selected from Hahl and Mitchell (1963). The dilute inflow is the weighted average inflow for the water year 1961 for the Bear River (site 39 of Hahl and Mitchell, 1963), and the lake composition is a single (representative) analysis of a water sample collected at site 123 just south of the railroad crossing in October 1960. The two water analyses are as follows:

[Data for Great Salt Lake, in millimoles per kilogram H_2O]

Constituent	*Inflow*	*Lake*
Ca	1.698	11.216
Mg	1.730	491.925
Na	9.449	5428.616
K	0.435	166.479
Cl	9.234	5834.640
SO_4	0.698	291.427
TDIC	6.021	6.304
pH	8.1	7.5

The original data were in parts per million. Because the lake water brine is too concentrated to be reliably speciated in WATEQFP, the two waters were speciated in PHRQPITZ (Plummer and others, 1988) to define the total dissolved inorganic carbon assuming a temperature of 15 °C. The mmolal concentrations from PHRQPITZ are given above and entered into DB. Although DB speciates the water through WATEQFP, the final total concentration of inorganic carbon is unaffected for data entered in molal (mmolal) units.

Saturation indices from PHRQPITZ indicate the lake water is saturated with halite, oversaturated with aragonite, slightly oversaturated with gypsum, and undersaturated with mirabilite. The calculated log P_{CO2} is -2.4 atmospheres, so there is potential for CO_2 outgassing. Spencer and others (1985a,b) note the presence of halite, mirabilite ($Na_2SO_4 \cdot 10H_2O$), and aragonite in lake sediment. Here NETPATH is used to solve for the extent of evaporation of the dilute inflow and the mass transfers of halite, mirabilite, aragonite, and CO_2 gas assuming a closed system with no further inputs.

The phases aragonite and mirabilite were not originally stored in the NETPATH.DAT file, but were generated using the <A>dd, phase command. In this problem aragonite is compositionally identical to calcite. The constraints Ca, Na, C, S, and Cl were included. The evaporation was selected using <E>dit. The default dissolution-only option for NaCl from the NETPATH.DAT file was edited, using <E>dit, Phase, to allow dissolution or precipitation. The main screen appears as follows:

[See file **NP-2.dat** on the diskette]

```
Initial Well:GSL, Bear R. weighted 1961
Final Well  :GSL, S. arm at RR Oct. 1960.
===============================================================================
     Constraints:  5      |       Phases:  4           |      Parameters
--------------------------|----------------------------|-----------------------
Carbon      Sulfur        | CO2 GAS  NaCl     ARAGONIT | Mixing: No
Calcium     Sodium        | MIRABILI                   | Evaporation: Yes
Chloride                  |                            | Rayleigh Calcs: No
===============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit
```

The input to NEWBAL and results are as follows:

```
Initial Well : GSL, Bear R. weighted 1961
Final Well   : GSL, S. arm at RR Oct. 1960.
         Final      Initial
C       6.3060       6.0210
S     291.4270        .6980
CA     11.2170       1.6980
NA   5428.6160       9.4490
CL   5834.6400       9.2340
Hit <Q> to quit, or <Enter> to continue

CO2 GAS C   1.0000RS  4.0000I1-25.0000I2100.0000
NaCl    NA  1.0000CL  1.0000
ARAGONITCA  1.0000C   1.0000RS  4.0000
MIRABILINA  2.0000S   1.0000RS  6.0000
Hit <Q> to quit, or <Enter> to continue

           1 models to be tested
           1 models were tested.
           1 models were found which satisfied the constraints.

           MODEL   1
 CO2 GAS           -3624.65663
 NaCl              -1897.19696
 ARAGONIT          -1410.55679
 MIRABILI           -293.02418
 Evaporation factor:    837.323
No more models, hit <Enter> to continue
```

The results indicate that 1 liter of Great Salt Lake water would have been evaporated from 837 liters of average inflow water and in so doing precipitated 1.9 moles of halite, 0.3 moles of mirabilite and 1.4 moles of aragonite. Accompanying the precipitation is a loss of 3.6 moles of CO_2 gas. It might be possible to check this model partially by calculating a mass balance for the lake-sediment system. Other complicating factors at Great Salt Lake are discussed by Spencer and others, (1985a, 1985b), and include historical variation in inflow composition, spatial differences in brine composition in different parts of the lake, additional brines trapped in pore fluids, outflow during paleo-high stands, input from geothermal sources, and diffusion of solutes into the sediment. These process too could be considered in NETPATH, but their evaluation would require additional data.

Using the evaporation factor of 837, conservative concentration of Mg and K in the average inflow would produce lake concentrations of 1.4 molal for Mg compared to the observed 0.49 molal; and 0.36 molal for K compared to the observed 0.17 molal concentration in the lake. Obviously, sinks for Mg and K are indicated. Spencer and others (1985a,b) attribute this loss of Mg and K to diffusion into the sediment pore fluid and incorporation into diagenetic minerals.

Additional modeling could take into account diagenetic sinks for K and Mg, and the possibility of redox reactions accompanying microbial oxidation of organic matter in the sediment.

Example 3: Origin of Sodium Bicarbonate Waters in the Atlantic Coastal Plain

Sodium-bicarbonate waters are very common in Atlantic and Gulf Coastal Plain sediments. For example, Foster (1950) reports ground water from the Coastal Plain of Mississippi containing more than 1500 mg/L HCO_3^- with high Na^+. The origin of this water type is generally attributed to dissolution of calcium carbonate and cation exchange of Ca^{2+} for Na^+ on marine clays. It has long been recognized that a source of CO_2 is required to react with calcium carbonate to form bicarbonate (Cederstrom, 1946; Foster, 1950), otherwise the dissolution-exchange reaction would produce sodium-carbonate water ($pH > 10.3$). The sodium bicarbonate waters typically have pH values near 8.4.

As the Coastal Plain ground-water systems become confined downgradient, an atmospheric or soil-zone source for the CO_2 is unlikely. The source of CO_2 has generally been attributed to organic material oxidized by microbial activity and has been a subject of considerable research lately (see for example Thorstenson and others, 1979; Chapelle and Knobel, 1985; Chapelle and others, 1987). Of particular interest and impact on the geochemical modeling of these waters is the actual electron-acceptor in the overall oxidation of the organic matter. The most obvious possibilites are (1) oxygen (system open to the atmosphere), (2) iron reduction, (3) sulfate reduction, and (4) methanogenesis.

This problem examines analyses of two waters from the Aquia aquifer of Maryland given by Chapelle and Knobel (1985). Three general compositional zones are recognized by Chapelle and Knobel (1985): an upgradient calcium magnesium bicarbonate-type water (zone I) evolving to a sodium calcium magnesium bicarbonate-type water (zone II) and then to a sodium bicarbonate-type water (zone III). This example considers the evolution of the zone I water (at well Fd-12) to the zone III water (at well Ff-35). The chemical data are given in the table below:

[Analytical data in milligrams per liter. Temperature of 17 °C assumed for well Fd-12]

Zone *Well Number*	*I* Fd-12	*III* Ff-35
Temperature	17	19
pH	7.6	8.4
Calcium	41	3
Magnesium	13	1.9
Sodium	3.8	140
Potassium	7.1	7.8
Bicarbonate	178	367
Chloride	1.5	2.4
Sulfate	19	13
Silica	14	11

Dissolved iron and hydrogen sulfide were not reported but are assumed to be very low (zero). For illustration purposes, only the carbon, calcium, and sodium data initially are considered. As additional processes are added, the constraints of redox, iron and sulfur will also be included.

The following models are considered:

Model (1). -- Calcite, CO_2, and Ca/Na exchange for the constraints Ca, Na, and C. This should identify the amount of CO_2 needed to make the zone III sodium bicarbonate-type water.

Model (2). -- Begin considering possible redox process for the CO_2 source identified in Model (1). Replace CO_2 with CH_2O and include O_2 gas among the phases and "redox" among the constraints. This is the case of aerobic oxidation of organic matter.

Model (3). -- Now replace the O_2 gas with goethite and a sink for Fe(II). Two possible sinks for Fe(II) are considered: Fe/Na ion exchange and magnetite. The waters are likely to be undersaturated with siderite, at least initially (plus, as shown below, some unrealistic redox results follow using siderite). Model (3) is a redox problem involving iron reduction, so the constraints of iron and redox need to be included. Note that both a source(s) and sink(s) of iron are needed because it is assumed that the Fe_T concentration is very low in both waters.

Model (4). -- This model builds on Model (3) by including sulfate reduction. Because dissolved iron and H_2S concentrations are low in the ground water, it is necessary to include sources and sinks for sulfur and iron. In this example, the iron sink in model (3) is replaced with pyrite and gypsum is included as a source. Although gypsum is not present in the sediments, the pore waters in the adjacent confining units have been shown to be calcium-sulfate waters near gypsum saturation (see for example Pucci and Owens, 1989). Inclusion of gypsum necessitates inclusion of sulfur among the constraints.

Model (5). -- This model investigates methanogenesis. Methanogenesis does not occur in the presence of dissloved sulfate (Lovley and Klug, 1986). Here, the sulfate reduction system (deleting iron and sulfur as constraints, and pyrite, gypsum and goethite as phases) is replaced with CH_4 gas while retaining the redox constraint and CH_2O as a phase.

Model (6). -- In Models (2)-(5) redox reactions considered the oxidation state of carbon in "organic matter" to be zero -- comparable to that of carbohydrates (CH_2O). It is likely that the source of the organic matter is lignite rather than carbohydrate. As discussed earlier in this report, the oxidation state of carbon in lignite is approximately -0.4. Model (6) repeats Model (4) (sulfate reduction) with lignite rather than carbohydrates.

The primary choice of phases included calcite, Ca/Na exchange and a carbon source. This differs somewhat from the treatment of Chapelle and Knobel (1985) who dissolved 10 mole percent Mg-calcite and precipitated 2 mole percent Mg-calcite. It is now known that calcites of only low Mg content and aragonite are present in the Paleocene clays, silts and sands of the Aquia Formation. For illustration purposes, reactions involving Mg have been ignored.

Model (1): Open to Carbon Dioxide Gas

The problem was set up as follows:

[See file **NP-3-1.dat** on the diskette]

```
Initial Well:Chapelle-Knobel (1985) Fd-12
Final Well  :Chapelle-Knobel (1985) Ff-35
=============================================================================
    Constraints:  3      |         Phases:  3          |      Parameters
-------------------------|-----------------------------|---------------------
Carbon       Calcium     | CALCITE  EXCHANGE CO2 GAS   | Mixing: No
Sodium                   |                             | Evaporation: No
                         |                             | Rayleigh Calcs: No
                         |                             | Exchange: Ca/Na
=============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit
```

"EXCHANGE" is a general phase in NETPATH.DAT for Ca-Mg/Na ion exchange. Pure Ca/Na exchange is the default value. Once EXCHANGE is selected as a phase, the Ca/Mg ratio in the exchange can be edited using the Edit menu. Alternatively, in this case, the phase Ca-Na Ex could have been chosen from NETPATH.DAT. Ca/Na exchange corresponds to the reaction

$$Na_2X + Ca^{2+} \rightarrow CaX + 2Na^{+} \quad .$$

The stoichiometric coefficients of the "phase" are defined as Na: 2, Ca: -1. Thus, if a positive mass transfer is calculated in NEWBAL, the reaction proceeds to the right, uptake of Ca and release of Na. Negative mass transfer for this phase would indicate the reaction proceeded to the left, that is, uptake of Na on the exchanger. In general, positive mass transfers calculated in NEWBAL indicate that the phase entered the aqueous solution, indicating dissolution (of a phase) or ingassing (of a gas). Negative mass transfers indicate precipitation or outgassing.

Model (1) is not a redox problem as defined, thus "redox" is not among the constraints. Mixing and Evaporation are not considered in this problem and the capability to make isotopic calculations is not selected.

Selecting <R>un lists the input to NEWBAL, and calculates the result.

```
Initial Well : Chapelle-Knobel (1985) Fd-12
Final Well   : Chapelle-Knobel (1985) Ff-35

         Final     Initial
C       5.9760      3.0740
CA       .0750      1.0230
NA      6.0930       .1650
Hit <Q> to quit, or <Enter> to continue

CALCITE CA  1.0000C   1.0000RS  4.0000I1   .0000I2   .0000
EXCHANGECA -1.0000NA  2.0000MG   .0000
CO2 GAS C   1.0000RS  4.0000I1-25.0000I2100.0000
Hit <Q> to quit, or <Enter> to continue
```

```
          1 models to be tested
          1 models were tested.
          1 models were found which satisfied the constraints.

          MODEL   1
 CALCITE               2.01600
 EXCHANGE              2.96400
 CO2 GAS                .88600
No more models, hit <Enter> to continue
```

The reaction is calcite dissolution, Ca for Na ion exchange (release of Na to solution), and ingassing of CO_2. Other models could be considered including mineral sources of Mg such as low Mg-calcite, Mg-montmorillonite dissolution, and Mg/Na exchange; but this obscures the redox points being demonstrated in the subsequent models developed for this example. Overall, Model (1) is basically the reaction sought, but it is not particularly realistic when considering the possible mechanisms for CO_2 entering into the aquifer. If the magnitude of the CO_2 gas mass transfer were on the order of 0 to 0.1 mmol/kg H_2O, it would be likely that the system is closed to CO_2, considering likely uncertainties in total carbon in the initial and (or) final waters. In the results above, the CO_2 term is large and a plausible source of CO_2 gas is needed. Chapelle and Knobel (1985) note that the ground water initially contains dissolved oxygen but becomes anoxic down gradient in the presence of lignitic material. Possible redox reactions are now considered.

Model (2): Aerobic Oxidation of Organic Matter

In considering the possibility of aerobic oxidation of organic matter, the phase CO_2 is replaced with "CH_2O" and O_2, and the constraint "redox" is added. The new problem looks like this:

[See file **NP-3-2.dat** on the diskette]

```
Initial Well:Chapelle-Knobel (1985) Fd-12
Final Well  :Chapelle-Knobel (1985) Ff-35
==============================================================================
    Constraints:  4        |        Phases:  4         |      Parameters
-------------------------- |---------------------------|----------------------
Carbon        Calcium      | CALCITE  EXCHANGE+O2 GAS  | Mixing: No
Sodium        Redox        |+"CH2O"                    | Evaporation: No
                           |                           | Rayleigh Calcs: No
                           |                           | Exchange: Ca/Na
==============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit
```

and the results are as follows:

```
Initial Well : Chapelle-Knobel (1985) Fd-12
Final Well   : Chapelle-Knobel (1985) Ff-35
         Final     Initial
C       5.9760      3.0740
CA       .0750      1.0230
NA      6.0930       .1650
RS     23.9040     12.2960
Hit <Q> to quit, or <Enter> to continue
```

```
CALCITE CA  1.0000C   1.0000RS  4.0000I1   .0000I2   .0000
EXCHANGECA -1.0000NA  2.0000MG   .0000
O2 GAS  RS  4.0000
"CH2O"  C   1.0000I1-25.0000I2   .0000
Hit <Q> to quit, or <Enter> to continue

           1 models to be tested
           1 models were tested.
           1 models were found which satisfied the constraints.

           MODEL   1
 CALCITE              2.01600
 EXCHANGE             2.96400
 O2 GAS    +           .88600
 "CH2O"    +           .88600
No more models, hit <Enter> to continue
```

This reaction could occur, but there would have to be a mechanism for introducing the 0.8 mmol of O_2 per kg of water. Chapelle and Knobel noted 5 to 10 mg/L dissolved oxygen (DO) in the recharge waters. These are typical values for the solubility of O_2 in water (about 8 mg/L at 25 °C). The maximum DO content of 10 mg/L of O_2 of molecular weight 32 corresponds to 0.3 mmol of O_2 per liter. Thus, nearly three times as much oxygen in recharge waters would have to enter the ground water for aerobic oxidation to account for CO_2 production. This could happen only in the unsaturated zone where the system is open to atmospheric oxygen. Chapelle and Knobel note that the sodium-bicarbonate waters only form in the deeper, confined, anoxic part of the Aquia. Therefore, the model is invalid because there is no plausible source of O_2; so other electron acceptors are considered.

Model (3): Ferric Iron Reduction

It is commonly observed in coastal plain sand aquifers, in proceeding from oxygenated recharge downgradient, that as the O_2 begins to vanish, there is an abrupt increase in dissolved ferrous ion. Lee and Strickland (1988) noted dissolved iron contents of 2 mg/L in Cretaceous coastal plain sediments in Georgia, and South Carolina. These values are large compared with background values of tens of micrograms per liter dissolved Fe in the oxygenated zone. The source of iron is presumably FeOOH. Our particular problem becomes a little unrealistic here because Chapelle and Knobel (1985) did not report dissolved iron data. It has been assumed that the dissolved iron concentration is zero. If all the reduced iron remained in solution, FeOOH alone would be included in the phases along with "redox" as a constraint. In our cases, because dissolved iron concentrations are low, both a source and sink of iron are needed. Even if 2 mg/L (0.04 mmol/kg H_2O) were reasonable for the iron reduction zone of the Aquia, this is not enough iron to account for the total electron transfer, so a sink of iron still would have to be proposed. The waters are probably undersaturated with siderite, so siderite was not considered as a sink. For example, sinks of Fe/Na ion exchange and magnetite (not a likely phase) were considered. The two cases of iron sinks were considered separately to avoid constructing even less likely models involving reactions between Fe/Na exchange and magnetite. Considering Fe(II)/Na ion exchange, the phase was created using <A>dd, Phase. The stoichiometry is Fe: 1, Na: -2, RS: +2. The model and results are as follows:

[See file **NP-3-3a.dat** on the diskette]

```
Initial Well:Chapelle-Knobel (1985) Fd-12
Final Well  :Chapelle-Knobel (1985) Ff-35
===============================================================================
     Constraints:  5       |          Phases:  5          |       Parameters
---------------------------|------------------------------|----------------------
Carbon        Calcium      | CALCITE  +"CH2O"  GOETHITE   | Mixing: No
Sodium        Redox        | FeII-Na   CO2 GAS            | Evaporation: No
Iron                       |                              | Rayleigh Calcs: No
                           |                              | Exchange: Ca/Na
===============================================================================
Warning: There is no data for Iron in 1 of the wells - zero will be used.
===============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit
```

Note that NETPATH prints a warning under the main screen. Because iron data were not entered in DB for these analyses, they are undefined. NETPATH assumes that the undefined concentrations in solution are zero and prints a warning. If this is not appropriate, it is necessary to return to DB and enter appropriate missing data.

```
Initial Well : Chapelle-Knobel (1985) Fd-12
Final Well   : Chapelle-Knobel (1985) Ff-35

         Final     Initial
C       5.9760      3.0740
CA       .0750      1.0230
NA      6.0930       .1650
RS     23.9040     12.2960
FE       .0000       .0000
Hit <Q> to quit, or <Enter> to continue

CALCITE CA  1.0000C   1.0000RS  4.0000I1   .0000I2   .0000
"CH2O"  C   1.0000I1-25.0000I2   .0000
GOETHITEFE  1.0000RS  3.0000
FeII-Na FE  1.0000NA -2.0000RS  2.0000
CO2 GAS C   1.0000RS  4.0000I1-25.0000I2100.0000
Hit <Q> to quit, or <Enter> to continue

           1 models to be tested
           1 models were tested.
           1 models were found which satisfied the constraints.

           MODEL   1
 CALCITE               -.94800
 "CH2O"    +            .74100
 GOETHITE              2.96400
 FeII-Na              -2.96400
 CO2 GAS               3.10900
No more models, hit <Enter> to continue
```

The above result indicates the reduction of nearly 3 mmols of ferric hydroxide with the Fe(II) replacing Na on the exchanger. CO_2 gas was included because an impossible reaction (precipitating organic matter) results if Na-Ca exchange is retained. The model requires an unrealistic and large source of CO_2 gas entering the ground water.

Next, another reduced iron sink, magnetite is considered. The average oxidation state of iron in magnetite is 8/3=2.67. Thus magnetite is more reduced than FeOOH (+3). The model is as follows:

[See file **NP-3-3b.dat** on the diskette]

```
Initial Well:Chapelle-Knobel (1985) Fd-12
Final Well  :Chapelle-Knobel (1985) Ff-35
================================================================================
     Constraints:  5      |        Phases:  5         |      Parameters
--------------------------|----------------------------|-----------------------
Carbon       Calcium      | CALCITE  EXCHANGE+"CH2O"   | Mixing: No
Sodium       Redox        | GOETHITE MAGNETIT          | Evaporation: No
Iron                      |                            | Rayleigh Calcs: No
                          |                            | Exchange: Ca/Na
================================================================================
Warning: There is no data for Iron in 1 of the wells - zero will be used.
================================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit

Initial Well : Chapelle-Knobel (1985) Fd-12
Final Well   : Chapelle-Knobel (1985) Ff-35

         Final     Initial
C       5.9760      3.0740
CA       .0750      1.0230
NA      6.0930       .1650
RS     23.9040     12.2960
FE       .0000       .0000
Hit <Q> to quit, or <Enter> to continue

CALCITE CA  1.0000C   1.0000RS  4.0000I1   .0000I2   .0000
EXCHANGECA -1.0000NA  2.0000MG   .0000
"CH2O"  C   1.0000I1-25.0000I2   .0000
GOETHITEFE  1.0000RS  3.0000
MAGNETITFE  3.0000RS  8.0000
Hit <Q> to quit, or <Enter> to continue

           1 models to be tested
           1 models were tested.
           1 models were found which satisfied the constraints.

           MODEL   1
 CALCITE              2.01600
 EXCHANGE             2.96400
 "CH2O"    +           .88600
 GOETHITE            10.63200
 MAGNETIT            -3.54400
No more models, hit <Enter> to continue
```

This type of reaction could occur, but there would have to be mineralogical evidence of a reduced iron phase in the system. There is no evidence for this reaction in the Aquia. It seems logical to conclude that in the absence of very high concentrations of dissolved iron, a different iron phase must be produced. Production of an iron sulfide mineral during sulfate reduction is a plausible mechanism.

Model (4): Combined Sulfate and Iron Reduction

It is commonly observed that the concentration of dissolved iron is very low during sulfate reduction because of the very low solubilities of iron sulfide phases. The models below consider pyrite as a product, recognizing that the phase "pyrite" can represent a range of semicrystalline FeS_2 phases. If sulfate reduction were occurring (in the absence of iron reactions), all the hydrogen sulfide produced would be found in solution. This would generate a noticable amount of H_2S considering the amount of CO_2 that must be produced. Therefore, if sulfate reduction is occurring in the Aquia aquifer, there must be a source and sink for sulfur and iron. The models below assume FeOOH for the source of iron. The question of the source of sulfate remains. This is particularly accute because there are no evaporite beds or nodules of gypsum in the formation. There is the possibility that SO_4 substitutes in the lattice of marine calcite (Busenberg and Plummer, 1985), but, because the calcites have recrystallized in freshwater, most of the sulfate has probably already been released during previous diagenesis. Recently Pucci and Owens (1989) observed calcium sulfate waters apporaching several thousand milligrams per liter in pore fluids of the confining units of Coastal Plain aquifers in New Jersey. McMahon and Chapelle (personal comm., 1990) considered the confining units as sources of sulfate in South Carolina. In the models below, $CaSO_4$ is included as a "phase". The problem could be more realistically modeled as mixing (diffusion) of a calcium sulfate water into the Aquia aquifer. The model is --

[See file **NP-3-4.dat** on the diskette]

```
Initial Well:Chapelle-Knobel (1985) Fd-12
Final Well  :Chapelle-Knobel (1985) Ff-35
==============================================================================
     Constraints:  6         |        Phases:  6          |      Parameters
-------------------------    |----------------------------|----------------------
Carbon       Calcium         | CALCITE  EXCHANGE+"CH2O"   | Mixing: No
Sodium       Redox           | GOETHITE+GYPSUM  -PYRITE   | Evaporation: No
Iron         Sulfur          |                            | Rayleigh Calcs: No
                             |                            | Exchange: Ca/Na
==============================================================================
Warning: There is no data for Iron in 1 of the wells - zero will be used.
==============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit

Initial Well : Chapelle-Knobel (1985) Fd-12
Final Well   : Chapelle-Knobel (1985) Ff-35

         Final     Initial
C       5.9760      3.0740
CA       .0750      1.0230
NA      6.0930       .1650
RS     24.7140     13.4840
FE       .0000       .0000
S        .1350       .1980
Hit <Q> to quit, or <Enter> to continue

CALCITE CA  1.0000C   1.0000RS  4.0000I1   .0000I2   .0000
EXCHANGECA -1.0000NA  2.0000MG   .0000
"CH2O"  C   1.0000I1-25.0000I2   .0000
GOETHITEFE  1.0000RS  3.0000
GYPSUM  CA  1.0000S   1.0000RS  6.0000I3 22.0000
PYRITE  FE  1.0000S   2.0000RS   .0000I3-60.0000
Hit <Q> to quit, or <Enter> to continue

            1 models to be tested
            1 models were tested.
            1 models were found which satisfied the constraints.
```

```
          MODEL   1
 CALCITE              1.13843
 EXCHANGE             2.96400
 "CH2O"    +          1.76357
 GOETHITE              .47029
 GYPSUM    +           .87757
 PYRITE    -          -.47029
No more models, hit <Enter> to continue
```

The above result is relatively realistic but should be investigated using carbon and sulfur isotope data (see Examples 5-7 below). For example if the sulfate of the confining unit were marine in origin (about 22 per mil CDT), the remaining sulfur in solution would be enriched due to microbial fractionation. It is interesting to note that so much calcium enters the solution with the sulfate that very little calcite dissolution is indicated. This model is considered in a later problem in demonstrating applications of the carbon isotope data.

Model (5): Methanogenesis

Fermentative bacteria metabolize complex organic matter to produce acetate, H_2, and CO_2, among other compounds. If sulfate is present, sulfate-reducers will metabolize H_2 and acetate to reduce sulfate. If sulfate is absent, methanogenic bacteria metabolize H_2 with acetate to produce methane. For methanogenesis with acetate, CO_2 and methane enter the solution simultaneously in 1:1 stoichiometric proportion, although they are not produced by the same metabolic process. The CO_2 is produced by fermentative bacteria and the methane by methanogens. (See Lovley and Klug, 1986, for details and further references).

This having been said, how is methanogenesis modeled for the Aquia aquifer? In the simplest case, consider an initial water as pure water and final water containing 1 mmol/L of total dissolved carbon (only). In addition to the carbon, there are no other elements other than hydrogen and oxygen in the final solution. This is a redox problem, so the two constraints are "carbon" and "redox". The two phases are CH_2O and CH_4.

The methane mass transfer into or out of the solution will depend on the redox state of the aqueous solution - that is, the oxidation state of carbon in the final solution. If methanogenesis had occurred according to the net reaction

$$2CH_2O \rightarrow CO_2 + CH_4 \quad ,$$

the 1 mmol of carbon in the final solution would be present as 0.5 mmol CO_2 and 0.5 mmol of CH_4. The redox state of this solution is

$$RS = 4m_{CO_2} - 4m_{CH_4} \quad ;$$

for this example of pure methanogenesis, RS is 0. Solving the carbon mass balance and redox balance equations, it can be shown that ΔCH_4 is 0.0 and ΔCH_2O is 1.0. This has sometimes been a point of confusion. Here, it was just concluded that methanogenesis was occurring, but a methane mass transfer of zero is calculated. In this case, methanogenesis surely did occur and the methane that was produced, 0.5 mmol, is in the solution. The redox state of the solution reflects this. NETPATH calculates that none of the methane that was produced has left the aqueous solution; thus, the methane mass transfer is zero.

Now, consider the possibility that the RS of the final solution is +4, and the solution contains 1 mmol of CO_2. Methanogenesis did again occur, but this time all the methane produced (1 mmol) has left the aqueous solution. In this case 2 mmols of "CH_2O" were consumed in methanogenesis producing 1 mmol of CO_2, which remains in the final solution, and 1 mmol of CH_4 which has left the aqueous solution, presumably through outgassing. With this in mind, the waters from the Aquia aquifer are reconsidered.

In the model below, carbohydrates and methane have been included among the phases, retaining "redox" as a constraint. The iron and sulfur constraints and phases are deleted (using <D>elete). The methanogenesis model is as follows:

[See file **NP-3-5.dat** on the diskette]

```
Initial Well:Chapelle-Knobel (1985) Fd-12
Final Well  :Chapelle-Knobel (1985) Ff-35
================================================================================
     Constraints:  4      |        Phases:  4           |      Parameters
--------------------------|-----------------------------|-----------------------
Carbon       Calcium      | CALCITE  EXCHANGE CH4 GAS   | Mixing: No
Sodium       Redox        |+"CH2O"                      | Evaporation: No
                          |                             | Rayleigh Calcs: No
                          |                             | Exchange: Ca/Na
================================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit

Initial Well : Chapelle-Knobel (1985) Fd-12
Final Well   : Chapelle-Knobel (1985) Ff-35

         Final     Initial
C       5.9760      3.0740
CA       .0750      1.0230
NA      6.0930       .1650
RS     23.9040     12.2960
Hit <Q> to quit, or <Enter> to continue

CALCITE CA  1.0000C   1.0000RS  4.0000I1   .0000I2   .0000
EXCHANGECA -1.0000NA  2.0000MG   .0000
CH4 GAS C   1.0000RS -4.0000I1-40.0000I2   .0000
"CH2O"  C   1.0000I1-25.0000I2   .0000
Hit <Q> to quit, or <Enter> to continue

           1 models to be tested
           1 models were tested.
           1 models were found which satisfied the constraints.

           MODEL   1
 CALCITE             2.01600
 EXCHANGE            2.96400
 CH4 GAS             -.88600
 "CH2O"    +         1.77200
No more models, hit <Enter> to continue
```

The results above show that if methanogenesis has occurred in the Aquia aquifer, all the methane produced (0.886 mmols/kg H_2O) has, somehow, left the system, because methane concentrations are very low in the Aquia aquifer (Chapelle and Knobel, 1985). Notice that there is exactly a 2:1 mass transfer ratio of organic matter:methane mass tranfer as dictated by the reaction stoichiometry.

Recalling our original definition of Redox State of aqueous solutions and its dependence on choice of constraints in the model, inspection of results from Models (4) and (5) shows different values of RS for the solutions. In model (4) RS is based on the dissolved C, Fe, and S data, and in model (5) RS is based only on the dissolved C data.

Model (6): Sulfate Reduction, Iron Reduction and Lignite

Recognizing that "CH_2O" may not be the most appropriate form of organic carbon source, the possibility of oxidation of lignite is considered. As discussed earlier, the oxidation state of carbon in lignite is approximately -0.4. This final model repeats Model (4), sulfate reduction, with a lignite carbon source.

[See file **NP-3-6.dat** on the diskette]

```
Initial Well:Chapelle-Knobel (1985) Fd-12
Final Well  :Chapelle-Knobel (1985) Ff-35
===============================================================================
     Constraints:  6      |         Phases:  6          |       Parameters
--------------------------|-----------------------------|----------------------
Carbon       Calcium      | CALCITE  EXCHANGE GOETHITE  | Mixing: No
Sodium       Redox        |+GYPSUM  -PYRITE  +LIGNITE   | Evaporation: No
Iron         Sulfur       |                             | Rayleigh Calcs: No
                          |                             | Exchange: Ca/Na
===============================================================================
Warning: There is no data for Iron in 1 of the wells - zero will be used.
===============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit

Initial Well : Chapelle-Knobel (1985) Fd-12
Final Well   : Chapelle-Knobel (1985) Ff-35

        Final     Initial
C       5.9760     3.0740
CA       .0750     1.0230
NA      6.0930      .1650
RS     24.7140    13.4840
FE       .0000      .0000
S        .1350      .1980
Hit <Q> to quit, or <Enter> to continue

CALCITE CA  1.0000C   1.0000RS  4.0000I1   .0000I2   .0000
EXCHANGECA -1.0000NA  2.0000MG   .0000
GOETHITEFE  1.0000RS  3.0000
GYPSUM  CA  1.0000S   1.0000RS  6.0000I3 22.0000
PYRITE  FE  1.0000S   2.0000RS   .0000I3-60.0000
LIGNITE C   1.0000RS  -.4000I1-25.0000I2   .0000
Hit <Q> to quit, or <Enter> to continue

           1 models to be tested
           1 models were tested.
           1 models were found which satisfied the constraints.
```

```
         MODEL   1
 CALCITE            .91087
 EXCHANGE          2.96400
 GOETHITE           .58406
 GYPSUM   +        1.10513
 PYRITE   -        -.58406
 LIGNITE  +        1.99113
No more models, hit <Enter> to continue
```

Note above in listing the input to NEWBAL, the RS for lignite is indicated as -0.4. The results using lignite are similar to those of Model (4) with "CH_2O", yet indicating increases in amounts of organic matter oxidation, goethite dissolution, sulfate reduction and pyrite formation, and small decreases in the amounts of calcite dissolved.

One final comment. All of the above models are possible until eliminated by introduction of additional data or observations, such as contradictions with saturation indices, inappropriate choice of phases, inconsistency with observed carbon and sulfur isotope data for the solution and solids, etc. Furthermore, the modeling is based on the assumption that the two chosen waters are truly on the same flow path in the Aquia aquifer, or are representative of waters on the same flow path.

Example 4: Mixing and Reaction -- Acid Mine Drainage at Pinal Creek, Near Globe, Arizona

The chemistry and migration of an acidic plume that has leaked from a waste pond used by a porphyry copper mining operation near Globe, Arizona, is currently being studied (Eychaner, 1989). The plume mixes in part with the natural background water in the basin-fill conglomerate. An overview of the hydrochemical setting is given in (Eychaner and others, 1989). Glynn (1991) made geochemical mass-balance models for the waters of the Pinal Creek site. The example below is an attempt to model the evolution of the final water in the plume at site 402 (pH 4-5) as a result of mixing of acidic water from the plume at site 51 ($pH < 4$) with the regional background water (site 404) plus reaction with the mineralogy of the basin-fill conglomerate. Rock types contributing to the conglomerate in the area include diabase, schist, granite, quartz monzonite, granodiorite, and quartzite. Limestones and dolostones are also present in the area. The three water analyses are as follows:

[Concentrations in milligrams per liter]

Parameter	*Site 404*	*Site 51*	*Site 402*
Temperature °C	19	18	18
pH	7.65	3.74	4.27
Dissolved O_2	5.9	.5	.6
Alk. or TDIC	[1]228	[2]254	[2]285
Calcium	40	440	540
Magnesium	13	390	190
Sodium	27	210	110
Iron	0	2,800	540
Manganese	.1	75	66
Chloride	8	340	140
Sulfate	18	8,800	3,300
Silica (SiO_2)	32	100	96
Aluminium	0	250	10

[1] *Total alkalinity as HCO_3^-.*
[2] *TDIC as HCO_3^-.*

In constructing the .PAT file, the measured Eh (0.43 volts) was used for sites 402 and 51. No Eh was available for site 404, so Eh was based on the Sato relation (see Plummer and others, 1976). In NETPATH, a mixing problem was set up where the two inital waters were from sites 404 and 51, and the final water was from site 402. The shallow unconfined system is considered open to CO_2 and O_2 gas. Models for this system are examined below in determining the mixing proportions of the two initial waters.

The following constraints were considered: C, S, Ca, Al, Mg, Na, Cl, Si, Fe, RS, and Mn. Chloride was considered to be conservative, so no chloride phases were included, and the mixing ratio was automatically based on Cl. Calcite, dolomite, and CO_2 were included for the carbonate system. There is a significant loss of dissolved iron in the plume, so the phase goethite was included. MnOOH was included to account for changes in dissolved Mn. To a first approximation, the alumino-silicate reactions were modeled using albite, SiO_2, gibbsite, and kaolinite. Clearly, many other silicate phases should ultimately be considered here, especially plagioclase feldspar(s). The system was considered open to O_2 gas and CO_2 gas. Organic matter is probably not present in significant quantities. The problem appears as follows in NETPATH:

[See file **NP-4.dat** on the diskette]

```
Initial Well:Pinal Cr. background, #404
Initial Well:Pinal Cr pH<4 plume, #51
Final Well  :Pinal Cr pH 4-5, #402
===============================================================================
     Constraints: 11      |        Phases: 11         |      Parameters
--------------------------|---------------------------|-----------------------
Carbon       Sulfur       |  CALCITE  CO2 GAS  GYPSUM   | Mixing: Yes
Calcium      Aluminum     |+ALBITE  +O2 GAS   GIBBSITE | Evaporation: No
Magnesium    Sodium       | GOETHITE+DOLOMITE MnOOH    | Rayleigh Calcs: No
Chloride     Silica       | KAOLINIT SiO2              |
Iron         Manganese    |                            |
Redox                     |                            |
===============================================================================
The mixing ratio will be determined by Chloride
===============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit

Initial Well 1 : Pinal Cr. background, #404
Initial Well 2 : Pinal Cr pH<4 plume, #51
Final Well     : Pinal Cr pH 4-5, #402

         Final    Initial 1   Initial 2
C       4.6960      3.9030      4.2200
S      34.5360       .1870     92.8760
CA     13.5450       .9980     11.1300
AL       .3730       .0000      9.3940
MG      7.8570       .5350     16.2640
NA      4.8100      1.1750      9.2610
CL      3.9700       .2260      9.7230
SI      1.6060       .5330      1.6870
FE      9.7210       .0000     50.8310
MN      1.2080       .0020      1.3840
RS    247.9390     17.4740    678.5860
Hit <Q> to quit, or <Enter> to continue

CALCITE CA  1.0000C   1.0000RS  4.0000I1   .0000I2   .0000
CO2 GAS C   1.0000RS  4.0000I1-25.0000I2100.0000
GYPSUM  CA  1.0000S   1.0000RS  6.0000I3 22.0000
ALBITE  NA  1.0000AL  1.0000SI  3.0000
O2 GAS  RS  4.0000
GIBBSITEAL  1.0000
GOETHITEFE  1.0000RS  3.0000
DOLOMITECA  1.0000MG  1.0000C   2.0000RS  8.0000I1   .0000I2   .0000
MnOOH   MN  1.0000RS  3.0000
KAOLINITAL  2.0000SI  2.0000
SiO2    SI  1.0000
Hit <Q> to quit, or <Enter> to continue

          11 models to be tested

          11 models were tested.
           3 models were found which satisfied the constraints.

Display models: <A>ll at once, [ENTER] for each, or <N>one.
```

```
           MODEL   1
 INIT 1    + F        .60577
 INIT 2    + F        .39423
 CALCITE             9.62327
 CO2 GAS           -11.19756
 GYPSUM             -2.19176
 ALBITE    +          .44726
 O2 GAS    +         2.32105
 GIBBSITE           -3.05394
 GOETHITE          -10.31809
 DOLOMITE  +         1.12116
 MnOOH                .66117
 KAOLINIT            -.36186
<Enter> to continue, any other key to quit

           MODEL   2
 INIT 1    + F        .60577
 INIT 2    + F        .39423
 CALCITE             9.62327
 CO2 GAS           -11.19756
 GYPSUM             -2.19176
 ALBITE    +          .44726
 O2 GAS    +         2.32105
 GIBBSITE           -3.77765
 GOETHITE          -10.31809
 DOLOMITE  +         1.12116
 MnOOH                .66117
 SiO2                -.72372
<Enter> to continue, any other key to quit

           MODEL   3
 INIT 1    + F        .60577
 INIT 2    + F        .39423
 CALCITE             9.62327
 CO2 GAS           -11.19756
 GYPSUM             -2.19176
 ALBITE    +          .44726
 O2 GAS    +         2.32105
 GOETHITE          -10.31809
 DOLOMITE  +         1.12116
 MnOOH                .66117
 KAOLINIT           -1.88883
 SiO2                3.05394
No more models, hit <Enter> to continue
```

For this set of phases, three models were found. The mixing fractions calculated from the chloride indicate that 61 percent of the final water is from the background basin-fill ground water (sample #404) and 39 percent is from the acid plume water (sample #51). The three models are similar, differing in the mass transfer of kaolinite, gibbsite, and silica. The predominate reactions appear to be calcite dissolution, outgassing of CO_2, precipitation of goethite, precipitation of gypsum, and ingassing of O_2 gas. It would be possible to check the mixing ratios on the basis of δD and $\delta^{18}O$ data, but these isotopic data are not currently available. It may be possible to check the carbonate reactions using carbon-isotope data.

Example 5: Application of Carbon Isotope Data to Example (3)

In Example (3), six models were solved for formation of sodium bicarbonate-type water in the Aquia aquifer of southern Maryland. The models considered different sources and sinks for carbon. Consequently, if the carbon isotopic composition of incoming carbon, and the fractionations when carbon leaves the aqueous phase (precipitation or outgassing) are known, it is possible to calculate values of $\delta^{13}C$ at the final well for comparison with the observed. Because a valid mass-balance model must predict the observed $\delta^{13}C$, models that fail to predict the observed $\delta^{13}C$ can be eliminated. Evaluation of the ^{13}C in the reaction is also a useful step in evaluating ^{14}C data and radiocarbon ages.

To solve this problem, it is first necessary to return to DB to add $\delta^{13}C$ values for TDIC for the initial and final waters in the Aquia aquifer from Chapelle and Knobel (1985). Although Chapelle and Knobel (1985) did not measure ^{14}C, it is instructive to include estimates of ^{14}C values for the initial and final waters to examine reaction effects on ^{14}C-age dating. So in DB the two wells Fd-12 and Ff-35 were edited to show the following $\delta^{13}C$ values for TDIC: -12.6 and -6.2 per mil (Chapelle and Knobel, 1985), respectively, and the following ^{14}C values: 50 and 2 percent modern (assumed), respectively. The revised .LON file was <S>aved through DB and a new .PAT file made.

Returning to NETPATH, the saved model file from Example 3, Model 6 (iron reduction coupled with sulfate reduction using lignite) is retrieved. This model is considered the most realistic for the Aquia aquifer. The retrieval is accomplished through the Edit menu. The listing below shows the retrieval of Model (6) from Example 3:

```
-----------------
CHOOSE MODEL FILE
-----------------
  1:  NP-1a.dat
  2:  NP-1b.dat
  3:  NP-1c.dat
  4:  NP-2.dat
  5:  NP-3-1.dat
  6:  NP-3-2.dat
  7:  NP-3-3a.dat
  8:  NP-3-3b.dat
  9:  NP-3-4.dat
 10:  NP-3-5.dat
 11:  NP-3-6.dat
 12:  NP-4.dat
 13:  NP-5-1.dat
 14:  NP-5-2.dat
 15:  NP-5-3.dat
 16:  NP-6.dat
 17:  NP-7.dat
Enter number of file, 0 for none, or <ENTER> to see more files:

11

---------------------------------------
NP-3-6.dat
---------------------------------------
Initial well: Chapelle-Knobel (1985) Fd-12
Final well  : Chapelle-Knobel (1985) Ff-35
CONSTRAINTS:
Carbon         Calcium        Sodium         Redox          Iron
Sulfur
PHASES:
CALCITE        EXCHANGE       GOETHITE       GYPSUM         PYRITE
LIGNITE
Use this file?  (N)o, (D)elete, or <ENTER> to accept:
```

```
Initial Well:Chapelle-Knobel (1985) Fd-12
Final Well  :Chapelle-Knobel (1985) Ff-35
==============================================================================
     Constraints:  6      |         Phases:  6          |      Parameters
--------------------------|-----------------------------|---------------------
Carbon        Calcium     | CALCITE  EXCHANGE GOETHITE  | Mixing: No
Sodium        Redox       |+GYPSUM  -PYRITE  +LIGNITE   | Evaporation: No
Iron          Sulfur      |                             | Rayleigh Calcs: No
                          |                             | Exchange: Ca/Na
==============================================================================
Warning: There is no data for Iron in 1 of the wells - zero will be used.
==============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit
```

Having retrieved Model (6) from Example 3, the Rayleigh calculations are invoked through the Edit menu. This is done by selecting <E>dit. Below is an example:

```
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit

e

 General
   1) Well file              : CLASS
   2) Entire model
   3) Phases
 Wells
   4) Mixing                 : No
   5) Initial well           : Chapelle-Knobel (1985) Fd-12
   6) Final well             : Chapelle-Knobel (1985) Ff-35
 Parameters
   7) Evaporation/Dilution   : No
   8) Ion exchange           : Ca/Na
   9) Redox state of DOC
 Isotope calculations
  10) Rayleigh calculations  : No
   -------------------
Edit which? (<Enter> when done)
10
Do Rayleigh calculations? (<Enter> for No )

y
```

Having selected the Rayleigh calculations, the Edit menu is expanded to allow entering and editing other isotopic data. In the example below, option 11 is selected to edit the isotopic data for carbon (and sulfur) phases in the model.

```
 General
   1) Well file               : CLASS
   2) Entire model
   3) Phases
 Wells
   4) Mixing                  : No
   5) Initial well            : Chapelle-Knobel (1985) Fd-12
   6) Final well              : Chapelle-Knobel (1985) Ff-35
 Parameters
   7) Evaporation/Dilution    : No
   8) Ion exchange            : Ca/Na
   9) Redox state of DOC
 Isotope calculations
  10) Rayleigh calculations   : Yes
  11) Isotopic data
  12) Model for initial C14   : Original Data
  13) Carbon fract. factors   : Mook
--------------------
Edit which? (<Enter> when done)

11
```

Chapelle and Knobel (1985) report the $\delta^{13}C$ for calcite (aragonite) from the Aquia aquifer to be +1.1 per mil. Organic matter in the Coastal Plain sediments is approximately -22 per mil in $\delta^{13}C$. It is assumed that ingassing of CO_2 is from fermentation of organic matter and, thus, also has a $\delta^{13}C$ of -22 per mil. All carbon sources within the aquifer are assumed to be old and their ^{14}C content is defined to be 0 pmc.

The sulfur isotopes are also treated in this problem, although the data are quite limited. The mass-transfer results are unaffected by lack of isotopic data because isotopic constratints were not included. In the Rayleigh calculations performed for each mass-balance model, it is assumed that incoming sulfate (presumably from pore spaces in the confining unit) is originally of marine origin having a $\delta^{34}S$ of 22 per mil. Since FeS_2 phases are sinks for iron and sulfur in the Aquia (that is, they do not dissolve in the system), it is not necessary to define the $\delta^{34}S$ of pyrite, but it is necessary to define the sulfur fractionation factor for precipitating pyrite. For example purposes it is assumed that pyrite is depleted in ^{34}S by 40 per mil relative to that of total dissolved sulfur in solution. The listing below shows the isotopic data for phases in the model.

```
                          Isotopic Compositions
Number  Phase       Carbon-13   C-14 (% mod)    Sulfur-34   Strontium-87
---------------------------------------------------------------------
    1:  CALCITE         .0000          .0000
    2:  GYPSUM                                    22.0000
    3:  PYRITE                                   -30.0000
    4:  LIGNITE      -25.0000          .0000
```

Every carbon-bearing phase in the model appears on the screen. NETPATH also considers Rayleigh processes with sulfur and strontium isotopes and phases containing these elements will also appear for editing. The default isotopic values for calcite, gypsum, pyrite, and lignite that are stored in the NETPATH.DAT file are displayed. The example below shows the results after editing the carbon isotopic values:

```
Enter number of phase to edit. <Enter> when done.
1

Enter isotopic composition of Carbon-13 for 'CALCITE '.
<Enter> to use    .0000
1.1
Enter isotopic composition of Carbon-14 for 'CALCITE '.
<Enter> to use    .0000

                            Isotopic Compositions
Number  Phase      Carbon-13   C-14 (% mod)    Sulfur-34   Strontium-87
--------------------------------------------------------------------
   1:  CALCITE        1.1000        .0000
   2:  GYPSUM                                    22.0000
   3:  PYRITE                                   -30.0000
   4:  LIGNITE     -25.0000         .0000
Enter number of phase to edit. <Enter> when done.

4

Enter isotopic composition of Carbon-13 for 'LIGNITE '.
<Enter> to use  -25.0000
-22
Enter isotopic composition of Carbon-14 for 'LIGNITE '.
<Enter> to use    .0000

                            Isotopic Compositions
Number  Phase      Carbon-13   C-14 (% mod)    Sulfur-34   Strontium-87
--------------------------------------------------------------------
   1:  CALCITE        1.1000        .0000
   2:  GYPSUM                                    22.0000
   3:  PYRITE                                   -30.0000
   4:  LIGNITE     -22.0000         .0000
Enter number of phase to edit. <Enter> when done.
```

None of the phases contain strontium, so there is no need to specify the $^{87}Sr/^{86}Sr$ ratio. Note that the isotopic values displayed for phases are used in NETPATH for any incoming source mineral. In this case, all dissolving calcite will enter the solution with $\delta^{13}C$ of +1.1 per mil and ^{14}C content of 0.0 pmc. These isotope values do not apply to precipitation which involves an isotopic fractionation process.

The next screen that appears under option 11 of the Edit menu allows editing of isotopic fractionation factors used in the Rayleigh calculations.

```
            Additive Fractionation Factors (in per mil)
                        Relative to solution
Number  Phase      Carbon-13     Carbon-14     Sulfur-34   Strontium-87
--------------------------------------------------------------------
   1:  CALCITE         .7048*       1.4096*
   2:  PYRITE                                  Undefined
* = based on computed value at 1.00 fraction between init and final waters
```

Note that all fractionation factors are given relative to the average isotopic composition of that element **in solution**. The "*" denotes values calculated using published equilibrium fractionation factors (either the Mook, 1980, or Deines and others, 1974 sets). In the case of pyrite the correlation from Plummer and others (1990) for the kinetic fractionation observed in the Madison aquifer is used. Use of the Madison aquifer correlation depends on data for $\delta^{34}S$ of sulfate (and temperature) in the final well (when the fraction X = 1.00). Because $\delta^{34}S$ of sulfate is not known in our final well, NETPATH shows the pyrite-solution sulfur fractionation as "undefined". For example purposes below, the pyrite fractionation is edited giving a value of -40. This means that any pyrite formed will be depleted in ^{34}S by 40 per mil relative to that of the aqueous solution. It is often instructive to examine the sensitivity of modeling results to uncertainties in isotopic data, including fractionation factors. The next test problem

(Example 6) is perhaps a more realistic application of sulfur isotope data. Because of the mass difference between ^{13}C and ^{14}C, it is theoretically predicted that ^{14}C frationation factors will be approximately twice those of ^{13}C (Craig, 1954). Alternatively, the ^{14}C-fractionation factors can be edited to user-defined values. The listing above shows the calculated and (or) default fractionation factors.

The "*" denotes fractionation factors that were calculated by NETPATH. They are given in per mil relative to the average isotopic composition of that element in solution. As shown in the Edit menu (above), the Mook equilibrium fractionation factors for the inorganic carbon system have been selected. At the current menu, the fraction between the initial and final water can be modified. It is also possible to modify fractionation factors. Because no data for $\delta^{34}S$ of sulfate were available at the fraction 1.00 between initial and final waters (that is, at the final well), the pyrite fractionation is undefined. By entering 0 here for the fraction between initial and final wells, fractionation factors can be recalculated for the conditions at the initial well (X=0.0), or at any linear combination between initial and final well. The fractionation factor can be edited by entering the phase number to edit. Below is an example of editing the pyrite fractionation factor to the suggested value of -40 per mil:

```
Enter number of phase to edit, '0' for fraction along path, <Enter> when done.
2
Enter additive fractionation factor of Sulfur-34 for 'PYRITE  '.
-40
             Additive Fractionation Factors (in per mil)
                        Relative to solution
Number  Phase      Carbon-13     Carbon-14    Sulfur-34   Strontium-87
----------------------------------------------------------------------
   1:  CALCITE          .7048*        1.4096*
   2:  PYRITE                                  -40.0000
* = based on computed value at 1.00 fraction between init and final waters
Enter number of phase to edit, '0' for fraction along path, <Enter> when done.
```

Once the selection of fractionation factors has been made, control is returned to the Edit menu. Because estimates of ^{14}C values for the initial and final wells are included, it is necessary to select option (12) from the Edit menu to define of the initial ^{14}C value. Here, models to define the initial ^{14}C value for the recharge water (prenuclear detonation A_0) are selected. These A_0 models are examined in more detail in Example 6. The present problem selects "original data" which causes NETPATH to set A_0 equal to the observed ^{14}C value of the initial well (defined as 50 pmc in DB). The listing below is an example of the selection procedure:

```
 General
   1) Well file                : CLASS
   2) Entire model
   3) Phases
 Wells
   4) Mixing                   : No
   5) Initial well             : Chapelle-Knobel (1985) Fd-12
   6) Final well               : Chapelle-Knobel (1985) Ff-35
 Parameters
   7) Evaporation/Dilution     : No
   8) Ion exchange             : Ca/Na
   9) Redox state of DOC
 Isotope calculations
  10) Rayleigh calculations    : Yes
  11) Isotopic data
  12) Model for initial C14    : Original Data
  13) Carbon fract. factors    : Mook
--------------------
Edit which? (<Enter> when done)
```

```
12
   Initial Carbon-14, A0, (percent modern)
         for Total Dissolved Carbon
         Model               Initial Well
  1 : Original Data        :    50.00
  2 : Mass Balance         :    55.76
  3 : Vogel                :    85.00
  4 : Tamers               :    52.76
  5 : Ingerson and Pearson:     50.40
  6 : Mook                 :    53.08
  7 : Fontes and Garnier   :    48.70
  8 : Eichinger            :    47.92
  9 : User-defined         :   100.00
Enter number of model to use (<Enter> for 'Original Data')
```

Note that the <Enter> choice was selected here. If another model were selected (other than the original data), the screen would prompt for data required for that particular A_0 model. Any other key would cause prompting for data for all A_0 models. The preliminary values for A_0 models are based on pre-set default values (See Example 6). Having provided the needed isotopic data, NETPATH returns to the Edit menu.

```
 General
   1) Well file               : CLASS
   2) Entire model
   3) Phases
 Wells
   4) Mixing                  : No
   5) Initial well            : Chapelle-Knobel (1985) Fd-12
   6) Final well              : Chapelle-Knobel (1985) Ff-35
 Parameters
   7) Evaporation/Dilution    : No
   8) Ion exchange            : Ca/Na
   9) Redox state of DOC
 Isotope calculations
  10) Rayleigh calculations   : Yes
  11) Isotopic data
  12) Model for initial C14   : Original Data
  13) Carbon fract. factors   : Mook
--------------------
Edit which? (<Enter> when done)
```

If satisfied with isotopic data and other parameters set through the Edit menu, <Enter> returns to the main screen and selecting <R>un solves the problem, as shown below.

[See file **NP-5-1.dat** on the diskette]

```
Initial Well:Chapelle-Knobel (1985) Fd-12
Final Well  :Chapelle-Knobel (1985) Ff-35
==========================================================================
     Constraints:  6        |        Phases:  6          |      Parameters
--------------------------- |--------------------------- |----------------------
Carbon        Calcium       | CALCITE  EXCHANGE GOETHITE | Mixing: No
Sodium        Redox         |+GYPSUM   -PYRITE  +LIGNITE | Evaporation: No
Iron          Sulfur        |                            | Rayleigh Calcs: Yes
                            |                            | Exchange: Ca/Na
                            |                            | Init C-14      50.00
                            |                            | (Original Data)
==========================================================================
Warning: There is no data for Iron in 1 of the wells - zero will be used.
==========================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit
```

```
r
Initial Well : Chapelle-Knobel (1985) Fd-12
Final Well   : Chapelle-Knobel (1985) Ff-35
        Final      Initial
C       5.9760      3.0740
CA       .0750      1.0230
NA      6.0930       .1650
RS     24.7140     13.4840
FE       .0000       .0000
S        .1350       .1980
Hit <Q> to quit, or <Enter> to continue

CALCITE CA  1.0000C   1.0000RS  4.0000I1  1.1000I2   .0000
EXCHANGECA -1.0000NA  2.0000MG   .0000
GOETHITEFE  1.0000RS  3.0000
GYPSUM  CA  1.0000S   1.0000RS  6.0000I3 22.0000
PYRITE  FE  1.0000S   2.0000RS   .0000I3-60.0000
LIGNITE C   1.0000RS  -.4000I1-22.0000I2   .0000
Hit <Q> to quit, or <Enter> to continue

           1 models to be tested
           1 models were tested.
           1 models were found which satisfied the constraints.

           MODEL   1
 CALCITE                .91087
 EXCHANGE              2.96400
 GOETHITE               .58406
 GYPSUM    +           1.10513
 PYRITE    -           -.58406
 LIGNITE   +           1.99113
               Computed    Observed
Carbon-13      -13.6438     -6.2000
C-14 (% mod)    25.7195*     2.0000
Sulfur-34         Insufficient data
Strontium-87      Insufficient data
-------------------------------------------------------------------------
Adjusted C-14 age in years:   21114*     * = Based on Original Data

No more models, <S> to show Rayleigh data, <C> to run all C-14 models,
<Enter> to continue
```

The computed mass transfer is, as expected, identical to that found previously in Model 6 of Example 3. But having included carbon-isotope data, the consistency of the model can be tested with the ^{13}C data (and other isotope data if they are available). The above result shows that the calculated $\delta^{13}C$ is far too light in comparison to that observed at the final well (-13.6 per mil calculated relative to the observed value of -6.2 per mil). It can be shown that, by editing the isotopic data, unreasonably heavy organic matter or calcite is required for the model to predict the observed $\delta^{13}C$. That is, the modeled $\delta^{13}C$ is inconsistent within the uncertainties in the known isotopic data. Therefore, the reaction model is very likely incorrect. Only two possible models could result in enriched ^{13}C values at the final well -- methanogenesis and carbon isotope exchange with the carbonate minerals. The calculations that follow consider the possibility of carbon isotope exchange (recrystallization of calcite). Returning to the main screen, the Edit menu is entered to edit the phase calcite to allow isotopic exchange of an estimated 5.2 mmol of calcite per kg H_2O. The listings below show this selection and the computed result.

```
Initial Well:Chapelle-Knobel (1985) Fd-12
Final Well  :Chapelle-Knobel (1985) Ff-35
==============================================================================
     Constraints:  6        |        Phases:  6           |      Parameters
--------------------------- |---------------------------- |---------------------
Carbon        Calcium       | CALCITE  EXCHANGE GOETHITE  | Mixing: No
Sodium        Redox         |+GYPSUM  -PYRITE  +LIGNITE   | Evaporation: No
Iron          Sulfur        |                             | Rayleigh Calcs: Yes
                            |                             | Exchange: Ca/Na
                            |                             | Init C-14      50.00
                            |                             | (Original Data)
==============================================================================
Warning: There is no data for Iron in 1 of the wells - zero will be used.
==============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit

e

 General
   1) Well file             : CLASS
   2) Entire model
   3) Phases
 Wells
   4) Mixing                : No
   5) Initial well          : Chapelle-Knobel (1985) Fd-12
   6) Final well            : Chapelle-Knobel (1985) Ff-35
 Parameters
   7) Evaporation/Dilution  : No
   8) Ion exchange          : Ca/Na
   9) Redox state of DOC
 Isotope calculations
  10) Rayleigh calculations : Yes
  11) Isotopic data
  12) Model for initial C14 : Original Data
  13) Carbon fract. factors : Mook
-------------------
Edit which? (<Enter> when done)
3

List of phases
  1:   CALCITE             2:   EXCHANGE             3:   GOETHITE
  4:   GYPSUM              5:   PYRITE               6:   LIGNITE
Input phase to edit.  <Enter to abort>
1

Input phase number (type 'L' to see phases and their corresponding numbers.)
75 to edit by piece, <Enter> for 'CALCITE '

Enter transfer allowed for 'CALCITE '. (1) dissolution, (2) precipitation,
(3) both, or (4) isotopic exchange.
Hit <Enter> for both.
4
Enter amount to exchange.  (0 to cancel).
5.2
Should 'CALCITE ' be included in every model?
<Enter> for no.
```

```
 General
   1) Well file              : CLASS
   2) Entire model
   3) Phases
 Wells
   4) Mixing                 : No
   5) Initial well           : Chapelle-Knobel (1985) Fd-12
   6) Final well             : Chapelle-Knobel (1985) Ff-35
 Parameters
   7) Evaporation/Dilution   : No
   8) Ion exchange           : Ca/Na
   9) Redox state of DOC
 Isotope calculations
  10) Rayleigh calculations  : Yes
  11) Isotopic data
  12) Model for initial C14  : Original Data
  13) Carbon fract. factors  : Mook
--------------------
Edit which? (<Enter> when done)
```

[See file **NP-5-2.dat** on the diskette]

```
Initial Well:Chapelle-Knobel (1985) Fd-12
Final Well  :Chapelle-Knobel (1985) Ff-35
===============================================================================
     Constraints:  6          |        Phases:  6          |     Parameters
--------------------------    |--------------------------- |----------------------
Carbon       Calcium          |*CALCITE  EXCHANGE GOETHITE | Mixing: No
Sodium       Redox            |+GYPSUM  -PYRITE  +LIGNITE  | Evaporation: No
Iron         Sulfur           |                            | Rayleigh Calcs: Yes
                              |                            | Exchange: Ca/Na
                              |                            | Init C-14      50.00
                              |                            | (Original Data)
===============================================================================
Warning: There is no data for Iron in 1 of the wells - zero will be used.
===============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit

r
Initial Well : Chapelle-Knobel (1985) Fd-12
Final Well   : Chapelle-Knobel (1985) Ff-35
         Final     Initial
C       5.9760     3.0740
CA       .0750     1.0230
NA      6.0930      .1650
RS     24.7140    13.4840
FE       .0000      .0000
S        .1350      .1980
Hit <Q> to quit, or <Enter> to continue

CALCITE CA  1.0000C   1.0000RS  4.0000I1  1.1000I2   .0000
EXCHANGECA -1.0000NA  2.0000MG   .0000
GOETHITEFE  1.0000RS  3.0000
GYPSUM  CA  1.0000S   1.0000RS  6.0000I3 22.0000
PYRITE  FE  1.0000S   2.0000RS   .0000I3-60.0000
LIGNITE C   1.0000RS  -.4000I1-22.0000I2   .0000
Hit <Q> to quit, or <Enter> to continue

           1 models to be tested
           1 models were tested.
           1 models were found which satisfied the constraints.
```

```
              MODEL   1
  CALCITE                 .91087         5.200 exchanged
  EXCHANGE               2.96400
  GOETHITE                .58406
  GYPSUM     +           1.10513
  PYRITE     -           -.58406
  LIGNITE    +           1.99113
                   Computed    Observed
Carbon-13          -6.2098     -6.2000
C-14 (% mod)        7.8021*     2.0000
Sulfur-34            Insufficient data
Strontium-87         Insufficient data
--------------------------------------------------------------------
Adjusted C-14 age in years:    11253*      * = Based on Original Data

No more models, <S> to show Rayleigh data, <C> to run all C-14 models,
<Enter> to continue
```

The amount of calcite exchanged, 5.2 mmol/kg H_2O, was determined by "trial and error", iterating between Edit Phase and <R>un. Notice that phases marked for isotopic exchange are identified with an "*" on the main screen, and the amount exchanged is displayed next to its calculated net mass transfer. In the above model, 0.91 mmol of calcite dissolved and an additional 5.2 mmol exchanged for every kilogram of H_2O along the flow path to the final well. In NETPATH "isotope exchange" or "recrystallization" refers to dissolution and subsequent precipitation of the same mass of the same phase. In the precipitation step, a fractionation occurs between precipitate and solution, and therefore, the Rayleigh calculations are used in NETPATH. In the case of recrystallization of calcite, a less stable calcite dissolves and is replaced by precipitation of another, more stable, calcite. In the dissolution/precipitation step, carbon from the mineral is exchanged with carbon in the aqueous solution. When the carbonates are significantly enriched relative to the solution, as for the marine carbonates in this problem, isotope exchange is a way of introducing more enriched carbon without changing the total concentration of calcium or carbon in solution (only the $\delta^{13}C$ of the solution is affected).

The results show that this particular model could be valid if there were substantial recrystallization of the carbonates. This is similar to the model that Chapelle and Knobel (1985) used, if aragonite were substituted for the 10 mol percent Mg-calcite reactant assumed by Chapelle and Knobel (1985). The above model could be modified by replacing reactant calcite with aragonite and including a low Mg-calcite as product. By changing the Mg mole fraction in low-Mg calcite, the recrystallization mass transfer changes. There may be some Mg content for the secondary calcite necessary to cause agreement in ^{13}C.

Notice also that recrystallization causes the modeled ^{14}C age to be substantially younger. There are obviously limits to the extent of isotopic exchange that could occur, taking into account the travel time for the water and the actual observed ^{14}C content of the water. The Rayleigh calculations were not carried out for the sulfur isotope data, as indicated by the statement "Insufficient data", even though the isotopic composition of sulfur-containing phases and the pyrite-solution fractionation factor were defined. In this case, the missing data are the sulfur isotopic composition of dissolved sulfate in the initial well and final well. Preliminary calculations assigning 22 per mil to the dissolved sulfate for the initial and final waters (entered through DB) show that the modeled isotopic composition of total dissolved sulfur (sulfate plus sulfide) at the final well (not shown here) would be substantially heavier than the pre-defined incoming sulfur from gypsum of 22 per mil (modeled $\delta^{34}S$ is 67 per mil at the final well). This is to be expected under conditions of sulfate reduction in systems of low dissolved sulfate content. Obviously, actual data on the sulfur isotopic composition of this system would be very useful in further refining the model.

Finally, the asymptotic $\delta^{13}C$ for the final solution is modeled for a large mass of isotopic exchange (Wigley and others, 1978). This is found by editing the mass of calcite exchanged to a very large amount, for example, 999.999 mmol/kg H_2O. The value 999.999 is the largest amount of exchange that can be saved in a model using the <S>ave option (although even larger values can be used within the model). The results are as follows:

[See file **NP-5-3.dat** on the diskette]

```
Initial Well : Chapelle-Knobel (1985) Fd-12
Final Well   : Chapelle-Knobel (1985) Ff-35
         Final     Initial
C       5.9760     3.0740
CA       .0750     1.0230
NA      6.0930      .1650
RS     24.7140    13.4840
FE       .0000      .0000
S        .1350      .1980
Hit <Q> to quit, or <Enter> to continue

CALCITE CA  1.0000C   1.0000RS  4.0000I1  1.1000I2   .0000
EXCHANGECA -1.0000NA  2.0000MG   .0000
GOETHITEFE  1.0000RS  3.0000
GYPSUM  CA  1.0000S   1.0000RS  6.0000I3 22.0000
PYRITE  FE  1.0000S   2.0000RS   .0000I3-60.0000
LIGNITE C   1.0000RS  -.4000I1-22.0000I2   .0000
Hit <Q> to quit, or <Enter> to continue

           1 models to be tested
           1 models were tested.
           1 models were found which satisfied the constraints.

           MODEL   1
 CALCITE                .91087      999.999 exchanged
 EXCHANGE              2.96400
 GOETHITE               .58406
 GYPSUM     +          1.10513
 PYRITE     -          -.58406
 LIGNITE    +          1.99113
              Computed    Observed
Carbon-13        .3511     -6.2000
C-14 (% mod)     .0000      2.0000
Sulfur-34        Insufficient data
Strontium-87     Insufficient data
No more models, <S> to show Rayleigh data, <C> to run all C-14 models,
<Enter> to continue
```

Thus, if a very large amount of calcite (aragonite) were involved in recrystallization, the $\delta^{13}C$ of the ground water would approach about +0.4 per mil (+0.3511 per mil calculated for 999.999 mmols of calcite exchanged) for equilibrium exchange conditions. It was not actually necessary to run NETPATH to find this out. Because the equilibrium isotopic fractionation factor for carbon between calcite and solution is 0.7048 per mil for the final well (as shown on the Fractionation Factor screen), it is expected that the final water will be depleted in ^{13}C by 0.7048 per mil relative to the calcite. The ^{13}C isotopic composition of calcite was defined to be +1.1 per mil, so the equilibrium $\delta^{13}C$ value for inorganic carbon in the final water is $+1.1 - 0.7048 = +0.3952$ per mil. This value is reached only after an infinite amount of recrystallization. Recrystallization of 999.999 mmol/kg H_2O results in a solution $\delta^{13}C$ value of +0.3511 per mil which is near the asymptotic value (+0.3952 per mil). For very large amounts of recrystallization, NETPATH approaches an asymptotic value of +0.3949 per mil for this problem. The small difference between the calculated and expected equilibrium value (0.0003 per mil) is attributed to uncertainties in numerical precision in the calculations.

Notice also that the ^{14}C content at the final well also places constraints on the extent of isotopic exchange (and other carbon-mass transfer) possible. If the value of 2-percent modern ^{14}C had actually been measured in the final well, the modeled ^{14}C content (0.0000-percent modern) is impossible. That is, an age younger than modern is implied. In this example, the extent of isotopic exchange must be less than that amount necessary to obtain an age no younger than modern for the water.

Example 6: Applications of ^{13}C, ^{14}C, and ^{34}S Data in Reaction Modeling, Madison Aquifer

Reactions in the Madison aquifer in parts of Wyoming, Montana and South Dakota were recently investigated by Plummer and others (1990). These authors showed that the predominant reactions were dedolomitization (calcite precipitation driven by dissolution of dolomite and anhydrite), sulfate reduction, iron reduction, cation exchange, and halite dissolution. Reaction models were found that reproduced the observed ^{13}C and ^{34}S at the final wells along eight flow paths. These reaction models were then applied to the initial ^{14}C data along the flow paths to estimate adjusted radiocarbon ages at the final wells.

In the example that follows, one reaction model from the Madison study is repeated to show combined uses of stable isotope data and examine the ^{14}C models for A_0. The chemical and isotopic data are taken from Plummer and others (1990) and summarized below for two waters, Recharge #3 and Mysse; Recharge #3 evolves to Mysse in south-central Montana.

[*Concentrations in millimoles per kilogram H_2O. n.d.-- not determined. Eh of Mysse well calculated from SO_4/H_2S data in WATEQFP. TDIC: total dissolved inorganic carbon calculated from pH and field alkalinity in WATEQFP*]

Constituent	*Recharge #3*	*Mysse*
Temperature (C)	9.9	63.
pH	7.55	6.61
Dissolved O_2	0.	0.
Eh (volts)	0.	-0.244
Ca^{2+}	1.2	11.28
Mg^{2+}	1.01	4.54
Na^{+}	0.02	31.89
K^{+}	0.02	2.54
Fe^{2+}	0.001	0.0004
Cl^{-}	0.02	17.85
SO_4^{2-}	0.16	19.86
H_2S_T	0.	0.259
TDIC	4.30	6.87
DOC	n.d.	n.d.
$\delta^{13}C$ *(TDIC)*	-6.99	-2.34
^{14}C *(TDIC) in pmc*	33.05	0.8
Tritium (TU)	19.8	n.d.
$\delta^{34}S_{SO4}$	9.73	16.30
$\delta^{34}S_{H2S}$	n.d.	-22.09

Other data used by Plummer and others (1990) define $\delta^{13}C$ of dolomite, calcite, and organic matter as +4.0, +4.0, and -25.0 per mil, respectively, and $\delta^{34}S$ of anhydrite and pyrite as +15.5 and -22.09 per mil, respectively. The ^{14}C content of all carbon sources along the flow path are 0.0 pmc. In defining A_0 using the mass-balance model, the $\delta^{13}C$ of soil gas CO_2 and calcite were taken to be -16.2 and +4.0 per mil (Plummer and others, 1990). Below is a listing of the main screen for this problem.

[See file **NP-6.dat** on the diskette]

```
Initial Well:Madison.Recharge #3
Final Well  :Madison.Mysse
============================================================================
     Constraints: 10       |          Phases: 10          |      Parameters
-------------------------- |----------------------------- |-----------------------
Carbon          Sulfur     |+DOLOMITE CALCITE +ANHYDRIT   | Mixing: No
Calcium         Magnesium  |+"CH2O"    GOETHITE-PYRITE    | Evaporation: No
Sodium          Potassium  | EXCHANGE+NaCl       SYLVITE  | Rayleigh Calcs: Yes
Chloride        Iron       | CO2 GAS                      | Exchange: Ca/Na
Redox           Sulfur-34  |                              | Init C-14       52.33
                           |                              | (Mass Balance)
============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit
```

Notice that the constraint Sulfur-34 was selected. In this case the sulfur isotope data will be treated as a mass-balance problem, even though there is a fractionating output of sulfur in precipitated pyrite. As discussed by Plummer and others (1990), the sulfur isotopic composition of anhydrite was adjusted to bring the CO_2-gas mass transfer to near zero, as expected for deep confined aquifers isolated from the atmosphere (that is, the system is expected to be closed to CO_2 gas). Before running the model, the Edit menu is used to view the isotopic data selected for the model.

```
e

 General
  1) Well file                : CLASS
  2) Entire model
  3) Phases
 Wells
  4) Mixing                   : No
  5) Initial well             : Madison.Recharge #3
  6) Final well               : Madison.Mysse
 Parameters
  7) Evaporation/Dilution     : No
  8) Ion exchange             : Ca/Na
  9) Redox state of DOC
 Isotope calculations
 10) Rayleigh calculations    : Yes
 11) Isotopic data
 12) Model for initial C14    : Mass Balance
 13) Carbon fract. factors    : Mook
--------------------
Edit which? (<Enter> when done)
```

```
11

                         Isotopic Compositions
Number  Phase      Carbon-13   C-14 (% mod)    Sulfur-34   Strontium-87
-----------------------------------------------------------------------
   1:  DOLOMITE      4.0000          .0000
   2:  CALCITE       4.0000          .0000
   3:  ANHYDRIT                                  15.5000
   4:  "CH2O"       -25.0000         .0000
   5:  PYRITE                                   -22.0900
   6:  CO2 GAS      -16.2000       100.0000
Enter number of phase to edit. <Enter> when done.

             Additive Fractionation Factors (in per mil)
                        Relative to solution
Number  Phase      Carbon-13     Carbon-14    Sulfur-34   Strontium-87
-----------------------------------------------------------------------
   1:  CALCITE        2.3814*       4.7629*
   2:  PYRITE                                   -37.8958@
   3:  CO2 GAS       -5.9231*      -11.8462*

* = based on computed value at .50 fraction between init and final waters
@ = based on computed value at final water
Enter number of phase to edit, '0' for fraction along path, <Enter> when done.
```

Notice that the carbon system fractionation factors are defined at the midpoint between the chemistries of the initial and final waters ($X=0.5$). This was changed from the default value of $X=1$.

The pyrite fractionation factor indicates that pyrite precipitated along the flow path is 37.8958 per mil lighter than the average isotopic composition of total dissolved sulfur in solution (SO_4+H_2S). This is calculated using the sulfur isotopic data at the Mysse well of $\delta^{34}S_{SO4}$ of +16.3 and $\delta^{34}S_{H2S}$ of -22.09

per mil. The total concentrations of sulfate and sulfide are 19.86 and 0.259 mmol. Thus, the average isotopic composition of S_T is 15.8058 per mil. Assuming that the pyrite will precipitate without further fractionation from the dissolved sulfide, the additive fractionation factor relative to the solution is

$$\epsilon_{pyr-soln} = \delta^{34}S_{H_2S} - \delta^{34}S_T = -22.09 - 15.8058 = -37.8958$$

If, however, the pyrite were fractionated by 1 per mil heavier than the dissolved sulfide, the pyrite solution fractionation factor could be changed to -36.8958.

The next screen pertains to various models for defining A_0 for ^{14}C calculaltions. These are accessed through the Edit menu is used.

```
 General
   1) Well file              : CLASS
   2) Entire model
   3) Phases
 Wells
   4) Mixing                 : No
   5) Initial well           : Madison.Recharge #3
   6) Final well             : Madison.Mysse
 Parameters
   7) Evaporation/Dilution   : No
   8) Ion exchange           : Ca/Na
   9) Redox state of DOC
 Isotope calculations
  10) Rayleigh calculations  : Yes
  11) Isotopic data
  12) Model for initial C14  : Mass Balance
  13) Carbon fract. factors  : Mook
--------------------
Edit which? (<Enter> when done)

12

   Initial Carbon-14, A0, (percent modern)
         for Total Dissolved Carbon
         Model               Initial Well
  1 : Original Data       :     33.05
  2 : Mass Balance        :     52.33
  3 : Vogel               :     85.00
  4 : Tamers              :     53.46
  5 : Ingerson and Pearson:     52.33
  6 : Mook                :     53.80
  7 : Fontes and Garnier  :     46.52
  8 : Eichinger           :     47.57
  9 : User-defined        :    100.00
Enter number of model to use (<Enter> for 'Mass Balance')

Hit <Enter> to input data for 'Mass Balance',
any other key to enter data for all models.

t
```

```
Enter value of C-14 activity in carbonate minerals (% modern),
<Enter> for       .00

Enter value of C-14 activity in soil gas CO2 (% modern),
<Enter> for    100.00

Enter value of C-14 activity in dolomite (% modern),
<Enter> for       .00

Enter value of C-14 activity in calcite (% modern),
<Enter> for       .00
Choices for delta C-13 (per mil) in the solution
 0 : Original Value
 1 : User-defined Value
Choose method for defining delta C-13 (per mil) in the solution,
<Enter> to use Original Value.

Enter value of delta C-13 (per mil) in carbonate minerals,
<Enter> for       .00
Choices for delta C-13 (per mil) in soil gas CO2
 0 : User-defined Value
 1 : Mass Balance - no fractionation
 2 : Mass Balance - with fractionation
 3 : Open System (gas-solution equilibrium)
Choose method for defining delta C-13 (per mil) in soil gas CO2,
<Enter> to use Mass Balance - no fractionation.

Enter value of delta C-13 (per mil) in dolomite,
<Enter> for       .00

Enter value of delta C-13 (per mil) in calcite,
<Enter> for       .00
C-13 of CO2 gas for initial well:  -13.359
Hit <Enter> to continue

   Initial Carbon-14, A0, (percent modern)
         for Total Dissolved Carbon
         Model              Initial Well
  1 : Original Data      :     33.05
  2 : Mass Balance       :     52.33
  3 : Vogel              :     85.00
  4 : Tamers             :     53.46
  5 : Ingerson and Pearson:    52.33
  6 : Mook               :     53.80
  7 : Fontes and Garnier :     46.52
  8 : Eichinger          :     47.57
  9 : User-defined       :    100.00
Enter number of model to use (<Enter> to quit, 0 to edit data for all models)

 General
   1) Well file               : CLASS
   2) Entire model
   3) Phases
 Wells
   4) Mixing                  : No
   5) Initial well            : Madison.Recharge #3
   6) Final well              : Madison.Mysse
 Parameters
   7) Evaporation/Dilution    : No
   8) Ion exchange            : Ca/Na
   9) Redox state of DOC
 Isotope calculations
  10) Rayleigh calculations   : Yes
  11) Isotopic data
  12) Model for initial C14   : Mass Balance
  13) Carbon fract. factors   : Mook
--------------------
Edit which? (<Enter> when done)
```

Returning to the main screen, the problem is run with the following result.

```
Initial Well : Madison.Recharge #3
Final Well   : Madison.Mysse
         Final     Initial
C       6.8700      4.3000
S      20.1190       .1600
CA     11.2800      1.2000
MG      4.5400      1.0100
NA     31.8900       .0200
K       2.5400       .0200
CL     17.8500       .0200
FE       .0000       .0010
RS    146.1220     18.1620
I3    317.9967      1.5568
Hit <Q> to quit, or <Enter> to continue

DOLOMITECA  1.0000MG  1.0000C   2.0000RS  8.0000I1  8.0000I2   .0000
CALCITE CA  1.0000C   1.0000RS  4.0000I1  4.0000I2   .0000
ANHYDRITCA  1.0000S   1.0000RS  6.0000I3 15.5000
"CH2O"  C   1.0000I1-25.0000I2   .0000
GOETHITEFE  1.0000RS  3.0000
PYRITE  FE  1.0000S   2.0000RS   .0000I3-44.1800
EXCHANGECA -1.0000NA  2.0000MG   .0000
NaCl    NA  1.0000CL  1.0000
SYLVITE K   1.0000CL  1.0000
CO2 GAS C   1.0000RS  4.0000I1-16.2000I2100.0000
Hit <Q> to quit, or <Enter> to continue

          1 models to be tested

          1 models were tested.
          1 models were found which satisfied the constraints.

          MODEL   1
 DOLOMITE +          3.53000
 CALCITE            -5.31723
 ANHYDRIT +         20.14723
 "CH2O"   +           .87067
 GOETHITE             .09311
 PYRITE   -          -.09411
 EXCHANGE            8.28000
 NaCl     +         15.31000
 SYLVITE             2.52000
 CO2 GAS             -.04345
              Computed    Observed
Carbon-13      -2.2140     -2.3400
C-14 (% mod)   12.2691*      .8000
Sulfur-34      15.8130     15.8058
Strontium-87     Insufficient data
-------------------------------------------------------------------
Adjusted C-14 age in years:   22570*     * = Based on Mass Balance
No more models, <S> to show Rayleigh data, <C> to run all C-14 models,
<Enter> to continue
```

Below, "S" is selected to show data used in the Rayleigh calculations and "C" to show radiocarbon ages calculated for the mass transfer model, but assuming different models for A_o.

```
s
Data used for Carbon-13
Initial Value:     -6.9900
 2 dissolving:
    DOLOMITE  Delta element:   7.06000  Isotopic composition:    4.0000
    "CH2O"    Delta element:    .87067  Isotopic composition:  -25.0000
 2 precipitating:
    CALCITE   Delta element:   5.31723  Fractionation factor:    2.3815
    CO2 GAS   Delta element:    .04345  Fractionation factor:   -5.9230
Data used for C-14 (% mod)
Initial Value:     52.3256
 2 dissolving:
    DOLOMITE  Delta element:   7.06000  Isotopic composition:-1000.0000
    "CH2O"    Delta element:    .87067  Isotopic composition:-1000.0000
 2 precipitating:
    CALCITE   Delta element:   5.31723  Fractionation factor:    4.7631
    CO2 GAS   Delta element:    .04345  Fractionation factor:  -11.8460
Data used for Sulfur-34
Initial Value:     9.7300
 1 dissolving:
    ANHYDRIT  Delta element:  20.14723  Isotopic composition:   15.5000
 1 precipitating:
    PYRITE    Delta element:    .18823  Fractionation factor:  -37.8958
Data used for Strontium-87
      Insufficient data
No more models, <S> to show Rayleigh data, <C> to run all C-14 models,
<Enter> to continue

c

 Model                      A0     Computed  Observed     Age
                                  (no decay)
-------------------------------------------------------------
Original Data              33.05      7.75        .80     18772
Mass Balance               52.33     12.27        .80     22570
Vogel                      85.00     19.93        .80     26580
Tamers                     53.46     12.54        .80     22748
Ingerson and Pearson       52.33     12.27        .80     22570
Mook                       53.80     12.62        .80     22800
Fontes and Garnier         46.52     10.91        .80     21598
Eichinger                  47.57     11.15        .80     21782
User-defined              100.00     23.45        .80     27924
No more models, <S> to show Rayleigh data, <C> to run all C-14 models,
<Enter> to continue
```

For this particular set of conditions, the A_0 values are quite similar in many of the models, and therefore similar ages are found, typically 21,000 to 23,000 years.

Example 7: Degradation of Dissolved Organic Carbon Released from Crude Oil

In the previous six examples, the carbon system has been limited to total dissolved inorganic carbon. In this problem, chemical and isotopic evolution in waters containing dissolved inorganic carbon, dissolved methane and dissolved organic carbon are investigated. The example is taken from a study of Baedecker and Cozzarelli (1991) on the degradation of organic carbon released from crude oil that contaminates a shallow sand aquifer near Bemidji, MN. Hult (1984) gives background data for the study. The uncontaminated glacial outwash material of the aquifer is primarily quartz sand with about 6 percent carbonate minerals (calcite), 30 percent feldspars, less than 5 percent clay minerals, and less than 0.2 percent organic carbon. Since 1979 when the oil spill occurred, contaminants have been transported more than 180 meters downgradient in a narrow plume. Beyond the plume, the DOC concentration in the ground water is less than 2 mg/L. Within the plume, the DOC concentration exceeds 50 mg/L. Baedecker and Cozzarelli (1991) modeled the degradation of the dissolved organic carbon released from the oil using a prerelease version of NETPATH and showed that the primary reactions appear to be (1) methanogenesis, (2) outgassing of CO_2 and CH_4 from the aquifer into the unsaturated zone, (3) dissolution of calcite and goethite, and (4) precipitation of a Ca-siderite ($Fe_{0.7}Ca_{0.3}CO_3$). Here the problem is repeated, similar to the original in which the gas mixture "CO_2-CH_4 is outgassed with a constant CO_2/CH_4 ratio equal to the ratio of the gas partial pressures measured in the plume, 0.12. This assumes that gas exchange in the unsaturated zone follows Henry's Law. The reaction is modeled between two water compositions in the plume; the upgradient water, Site A, near the oil body, and Site B about 40 meters downgradient. Listed below are the chemical and isotope data for the water at Sites A and B:

[Data of Baedecker and Cozzarelli (1991). Concentrations in milligrams per liter]

Property or Constituent	*Site A*	*Site B*
Temperature °C	10	9
pH	6.78	6.93
Eh (volts)	-.15	-.09
Dissolved O_2	.0	.0
Ca^{2+}	133.	122.
Mg^{2+}	43.0	33.2
Na^{+}	3.1	2.1
K^{+}	6.6	1.4
Sr^{2+}	.2	.1
Fe^{2+}	58.9	4.41
Mn^{2+}	.9	7.0
Cl^{-}	.3	.3
SO_4^{2-}	.048	1.19
H_2S	.025	.0
HCO_3^{-}	735.	495.
DOC (as C)	50.1	18.2
$CH_4(aq)$	21.7	6.2
NO_3^{-} *(as N)*	.5	.0
NH_4^{+} *(as N)*	.4	.076
SiO_2	61.1	30.5
$\delta^{13}C_{TDIC}$ *(per mil)*	-5.8	-5.95
$\delta^{13}C_{CH4(aq)}$	-56.1	-53.6

Other data needed to model the reaction are unknown and need to be estimated. These include (1) the mole fraction of Ca in siderite (assumed below to be 0.3), (2) the $\delta^{13}C$ of the DOC (assumed to be -25 per mil), (3) the oxidation state of carbon in the DOC (assumed to be -1, consistent with benzene, C_6H_6, as the predominant hydrocarbon released from the oil and subsequently degraded via microbial activity), and (4) the $\delta^{13}C$ of the calcite (assumed to be +1.5 per mil, which is reasonable for marine carbonates that occur in the glacial outwash deposits). Alternatively, other outgassing mechanisms could be considered that could affect the CO_2/CH_4 ratio in the gas mixture (assumed to be 0.12 here). The model that follows is nonunique, but does show that for reasonable values of unknown parameters, reactions can be found that are consistent with the chemical and isotopic data. The following is a listing of the main screen for this problem, critical data used (displayed from the Edit menu), and the modeled result:

[See file **NP-7.dat** on the diskette]

```
Initial Well:Bemidji Site A (well 421)
Final Well  :Bemidji Site B (well 533b-87)
================================================================================
    Constraints:  6         |          Phases:  6          |       Parameters
--------------------------- | ---------------------------- | ----------------------
Carbon        Calcium       | CALCITE  MnO2     NH4/CAEX   | Mixing: No
Iron          Manganese     | CO2-CH4  GOETHITE CA-FECO3   | Evaporation: No
Nitrogen      Redox         |                              | Rayleigh Calcs: Yes
                            |                              | X CO2 in CO2-CH4:  .12
                            |                              | Init C-14         .00
                            |                              | (Original Data)
================================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit


e

 General
   1) Well file              : CLASS
   2) Entire model
   3) Phases
 Wells
   4) Mixing                 : No
   5) Initial well           : Bemidji Site A (well 421)
   6) Final well             : Bemidji Site B (well 533b-87)
 Parameters
   7) Evaporation/Dilution   : No
   8) Fraction CO2 in CO2-CH4 :  .120
   9) Redox state of DOC
 Isotope calculations
  10) Rayleigh calculations  : Yes
  11) Isotopic data
  12) Model for initial C14  : Original Data
  13) Carbon fract. factors  : Mook
--------------------
Edit which? (<Enter> when done)

9
```

In selecting option (9), the redox state of the DOC reacting in the ground water is changed from the default value of 0.0 originally stored in the .PAT file from DB to -1.0 for benzene. Baedecker and Cozzarelli (1991) found a range of aromatic hydrocarbons released to the the ground water from the oil, including benzene (C_6H_6) and the alkylbenzenes (C_6 to C_{10}). Eganhouse (1987) and Cozzarelli and others (1990) describe the microbial degradation of the aromatic hydrocarbons at Bemidji. Further analytical studies of the distribution of organic contaminants and their relative rates of degradation in the plume are needed to refine the estimate of the redox state of the DOC.

```
The orginal value for RS of DOC in Bemidji Site A (well 421)
 is      .00. Do you want to enter a new value? (<Enter> = no)

y

Enter value for redox state of DOC for Bemidji Site A (well 421)

-1

The orginal value for RS of DOC in Bemidji Site B (well 533b-87)
 is      .00. Do you want to enter a new value? (<Enter> = no)

y

Enter value for redox state of DOC for Bemidji Site B (well 533b-87)

-1

 General
   1) Well file              : CLASS
   2) Entire model
   3) Phases
 Wells
   4) Mixing                 : No
   5) Initial well           : Bemidji Site A (well 421)
   6) Final well             : Bemidji Site B (well 533b-87)
 Parameters
   7) Evaporation/Dilution   : No
   8) Fraction CO2 in CO2-CH4 :  .120
   9) Redox state of DOC
 Isotope calculations
  10) Rayleigh calculations  : Yes
  11) Isotopic data
  12) Model for initial C14  : Original Data
  13) Carbon fract. factors  : Mook
--------------------
Edit which? (<Enter> when done)


11

                        Isotopic Compositions
Number  Phase      Carbon-13   C-14 (% mod)    Sulfur-34   Strontium-87
-----------------------------------------------------------------------
   1:  CALCITE        1.5000        .0000
   2:  CO2          -21.4500        .0000
       CH4          -53.6000        .0000
   3:  CA-FECO3        .0000        .0000
Enter number of phase to edit. <Enter> when done.
```

Although realistic ^{13}C values are given for CO_2 and CH_4 in the gas mixture, and for the Ca-siderite, the calculations are independent of these values because the mass transfers for these phases are, in this case, all negative, that is, they outgas or precipitate. Therefore, their isotopic compositions are determined by isotopic fractionation. Only the isotopic composition of calcite, which dissolves, is used in the following model:

```
                  Isotopic compositions of Carbon in solution
                                        Carbon-13  C14 %mod   Carbon-13  C14 %mod
#  Well Name                             of CH4     of CH4     of DOC     of DOC
--------------------------------------------------------------------------------
1: Bemidji Site A (well 421)            -56.100      .000    -25.000      .000
2: Bemidji Site B (well 533b-87)        -53.600      .000    -25.000      .000
Enter number of well to change, <Enter> when done.
```

The ^{13}C (and ^{14}C) values for dissolved methane and DOC are not commonly measured, therefore, they are not entered in DB. However, in considering carbon isotopic effects in modeling reactions with waters containing these forms of carbon, it is necesary to know the carbon isotopic compositions of all carbon species to define the isotopic composition of total dissolved carbon. The above screen only appears for initial and (or) final waters that contain either dissolved methane and (or) dissolved organic carbon. In the calculations that follow the isotopic composition of dissolved methane was determined by Baedecker and Cozzarelli (1991) and that of the DOC was assumed.

```
              Additive Fractionation Factors (in per mil)
                          Relative to solution
Number  Phase      Carbon-13     Carbon-14    Sulfur-34   Strontium-87
-----------------------------------------------------------------------
   1:  CALCITE        6.4680*        .0000
   2:  CO2           -3.4152*        .0000
       CH4          -43.9000         .0000
   3:  CA-FECO3       6.4680*        .0000
* = based on computed value at 1.00 fraction between init and final waters
Enter number of phase to edit, '0' for fraction along path, <Enter> when done.
```

The above screen defines the fractionation factors for all carbon-bearing phases that can occur in the model. These fractionation factors are used only for precipitation or outgassing. Because calcite actually dissolves, the calculations do not depend on the calcite fractionation factor. Note that all fractionation factors are calculated or defined relative to the average isotopic composition of the element in aqueous solution, which in this case is -9.74 per mil for total dissolved carbon at the final well ($X=1$). The methane fractionation factor between $CH_{4(aq)}$ and $CH_{4(gas)}$ was assumed to be zero. Therefore, the $\delta^{13}C$ composition of $CH_{4(gas)} = \delta^{13}C$ of $CH_{4(aq)} = -53.6$ per mil. And the fractionation factor for methane gas relative to the ^{13}C of the total dissolved carbon is $-53.6 - (-9.74) = -43.9$ per mil. In outgassing of the CO_2-CH_4 gas mixture, the two gases are assumed to leave the aqueous solution independently according to their mole fractions in the mixture. Therefore, the average fractionation factor for the gas mixture is computed as a linear proportion of the end-member fractionation factors. In this case, ϵ for the gas mixture containing 12 percent CO_2 is $0.12(-3.4152) + 0.88(-43.9) = -39.04$ per mil relative to the solution. The problem could be further refined by study of the siderite fractionation factor. Here the siderite fractionation factor was assumed equal to that of calcite. Next the main screen is invoked to run NETPATH.

```
 General
   1) Well file              : CLASS
   2) Entire model
   3) Phases
 Wells
   4) Mixing                 : No
   5) Initial well           : Bemidji Site A (well 421)
   6) Final well             : Bemidji Site B (well 533b-87)
 Parameters
   7) Evaporation/Dilution   : No
   8) Fraction CO2 in CO2-CH4 :  .120
   9) Redox state of DOC
 Isotope calculations
  10) Rayleigh calculations  : Yes
  11) Isotopic data
  12) Model for initial C14  : Original Data
  13) Carbon fract. factors  : Mook
--------------------
Edit which? (<Enter> when done)
```

```
Initial Well:Bemidji Site A (well 421)
Final Well  :Bemidji Site B (well 533b-87)
===============================================================================
     Constraints:  6     |        Phases:  6           |     Parameters
-------------------------|----------------------------|----------------------
Carbon       Calcium     | CALCITE  MnO2     NH4/CAEX | Mixing: No
Iron         Manganese   | CO2-CH4  GOETHITE CA-FECO3 | Evaporation: No
Nitrogen     Redox       |                            | Rayleigh Calcs: Yes
                         |                            | X CO2 in CO2-CH4:  .12
                         |                            | Init C-14        .00
                         |                            | (Original Data)
===============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit


r

Initial Well : Bemidji Site A (well 421)
Final Well   : Bemidji Site B (well 533b-87)
         Final     Initial
C      12.4670    22.5300
CA      3.0460     3.3220
FE       .0790     1.0560
MN       .1280      .0160
N        .0050      .0640
RS     39.6000    60.6780
Hit <Q> to quit, or <Enter> to continue

CALCITE CA  1.0000C   1.0000RS  4.0000I1  1.5000I2   .0000
MnO2    MN  1.0000RS  4.0000
NH4/CAEXN  +2.0000CA -1.0000RS -6.0000
CO2-CH4 C   1.0000RS -3.0400I1-49.7420I2   .0000
GOETHITEFE  1.0000RS  3.0000
CA-FECO3CA   .3000FE   .7000C   1.0000RS  5.4000I1   .0000I2   .0000
Hit <Q> to quit, or <Enter> to continue

           1 models to be tested

           1 models were tested.
           1 models were found which satisfied the constraints.

           MODEL   1
 CALCITE              3.04451
 MnO2                  .11200
 NH4/CAEX             -.02950
 CO2-CH4             -1.94081
 GOETHITE             6.83969
 CA-FECO3           -11.16670
               Computed    Observed
Carbon-13       -9.9119     -9.7398
DIC C-13        -6.1530     -5.9500
C-14 (% mod)      Insufficient data
Sulfur-34         Insufficient data
Strontium-87      Insufficient data

No more models, <S> to show Rayleigh data, <C> to run all C-14 models,
<Enter> to continue
```

The calculations require dissolution of approximately 3 mmols of calcite, 0.1 mmols of MnO_2, and 6.8 mmol of goethite, precipitation of 11.2 mmol of $Fe_{0.7}Ca_{0.3}CO_3$, and outgassing of 2 mmols of a CO_2-CH_4 gas mixture containing 12 mole percent CO_2 to model the observed ^{13}C at Site B. Notice that when DOC and (or) CH_4 are included, NETPATH prints the modeled and observed ^{13}C of TDC and of TDIC. Below, the Rayleigh data are displayed by entering <S> showing that NETPATH used, in this case, the fractionation factor of -39.04 per mil for the CO_2-CH_4 gas mixture, as discussed above. No data for ^{14}C, ^{34}S, or $^{87}Sr/^{86}Sr$ were available for this study.

```
s

Data used for Carbon-13
Initial Value:     -12.3752
 1 dissolving:
    CALCITE   Delta element:   3.05357  Isotopic composition:    1.5000
 2 precipitating:
    CO2-CH4   Delta element:   1.97467  Fractionation factor:  -39.0418
    CA-FECO3  Delta element:  11.14190  Fractionation factor:    6.4680
Data used for C-14 (% mod)
      Insufficient data
Data used for Sulfur-34
      Insufficient data
Data used for Strontium-87
      Insufficient data
No more models, <S> to show Rayleigh data, <C> to run all C-14 models,
<Enter> to continue

Initial Well:Bemidji Site A (well 421)
Final Well  :Bemidji Site B (well 533b-87)
==============================================================================
     Constraints:  6         |        Phases:  6           |      Parameters
-------------------------|----------------------------|----------------------
Carbon       Calcium         | CALCITE  MnO2       NH4/CAEX | Mixing: No
Iron         Manganese       | CO2-CH4  GOETHITE CA-FECO3 | Evaporation: No
Nitrogen     Redox           |                            | Rayleigh Calcs: Yes
                             |                            | X CO2 in CO2-CH4:  .12
                             |                            | Init C-14         .00
                             |                            | (Original Data)
==============================================================================
Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or <Q>uit
```

At this point, the user could continue with this problem using NETPATH to examine the sensitivity of modeling results to uncertainties in the compositional and isotopic data. Such analysis helps focus the investigation on critical data needed to understand further the reaction system. In this problem, the most significant data needs appear to be the redox state of the reacting DOC, the calcium content of the siderite, and the isotopic composition of the dissolving calcite. Other studies should investigate the mechanism of CO_2-CH_4 outgassing and its effect on gas composition. Furthermore, because this problem deals with a point source of contamination, it may be necessary to evaluate the effect of hydrodynamic dispersion on the water chemistry. This could be accomplished through digital simulation of the flow and solute transport.

REFERENCES

Baedecker, M.J., and Cozzarelli, I.M., 1991, Geochemical modeling of organic degradation reactions in an aquifer contaminated with crude oil: U.S. Geological Survey Toxic Substances Hydrology Program--Proceedings of the Technical Meeting, Monterey California, March 11-15, 1991, Mallard, G.E., and Aronson, D.A., Eds., Water-Resources Investigations Report 91-4034, [in press].

Busenberg, E., and Plummer, L.N., 1985, Kinetic and thermodynamic factors controlling the distribution of SO_4^{2-} and Na^+ in calcites and selected aragonites: Geochimica et Cosmochimica Acta, v. 49, p. 713-725.

Cederstrom, D.J., 1946, Genesis of ground waters in the Coastal Plain of Virginia: Economic Geology, v. 41, p. 218-245.

Chapelle, F.H., and Knobel, L.L., 1985, Stable carbon isotopes of HCO_3- in the Aquia aquifer, Maryland: Evidence for an isotopically heavy source if CO_2: Ground Water, v. 23, no. 5, p. 592-599.

Chapelle, F.H., Zelibor, J.L., Jr., Grimes, D.J., and Knobel, L.L., 1987, Bacteria in deep coastal plain sediments of Maryland: A possible source of CO_2 to groundwater: Water Resources Research, v. 23, no. 8, p. 1625-1632.

Chapelle, F.H., Morris, J.T., McMahon, P.B., and Zelibor, J.L., Jr., 1988, Bacterial metabolism and the $\delta^{13}C$ composition of ground water, Floridan aquifer system, South Carolina: Geology, v. 16, p. 117-121.

Chen, S., and Long, A., 1984, Implementation of carbon isotope subroutine to computer program PHREEQE and their application to C-14 ground-water dating: Proceedings, Hydrology and Water Resources in Arizona and the Southwest: v. 14, American Water Resources Association, Tucson, Arizona, p. 121-135.

Claypool, G.E., and Kaplan, I.R., 1974, The origin and distribution of methane in marine sediments:, *in* Kaplan, I.R., ed., Natural Gases in Marine Sediments: New York, N.Y., Plenum Press, p. 99-139.

Claypool, G.E., Holser, W.T., Kaplan, I.R., Sakai, H., and Zak, I., 1980, The age curves of sulfur and oxygen isotopes in marine sulfate and their mutual interpretation: Chemical Geology, v. 27, p. 199-260.

Cozzarelli, I.M., Eganhouse, R.P., and Baedecker, M.J., 1990, Transformaiton of monoaromatic hydrocarbons to organic acids in anoxic groundwater environment: Environmental Geology and Water Science, v. 16, p. 135-141.

Craig, H., 1954, Carbon-13 in plants and the relationships between carbon-13 and carbon-14 in nature: Journal of Geology, v. 62, p. 115-149.

Deines, P., Langmuir, D., and Harmon, R.S., 1974, Stable carbon isotope ratios and the existence of a gas phase in the evolution of carbonate ground waters: Geochimica et Cosmochimica Acta, v. 38, p. 1147-1164.

Eganhouse, R.P., Dorsey, T.F., and Phinney, C.S., 1987, Transport and fate of monoaromatic hydrocarbons in the subsurface, Bemidjii, Minnesota, research site, *in* Franks, B.J., ed., U.S. Geological Survey Program on Toxic Waste--Ground-Water contamination: Proceedings of the Third Technical Meeting, Pensacola, Florida, 1987: U.S. Geological Survey Open-File Report 87-109, p. C29.

Eichinger, L., 1983, A contribution to the interpretation of ^{14}C groundwater ages considering the example of a partially confined sandstone aquifer: Radiocarbon, v.25, p. 347-356.

Eychaner, J.H., 1989, Movement of inorganic contaminants in acidic water near Globe, Arizona, *in* Mallard, G.E., and Ragone, S.E., eds., U.S. Geological Survey Toxic Substances Hydrology Program--Proceedings of the Technical Meeting, Phoenix, Arizona, Sept. 26-30, 1988: U.S. Geological Survey Water-Resources Investigations Report 88-4220, p. 567-575.

Eychaner, J.H., Rehmann, M.R., and Brown, J.G., 1989, Chemical, geologic and hydrologic data from the study of acidic contamination in the Miami Wash-Pinal Creek area, Arizona, water years 1984-87: U.S. Geological Survey Open-File Report 90-395, 105 p.

Faure, G., 1986, Principles of Isotope Geology: New York, N.Y., John Wiley & Sons, 589 p.

Feth, J.H., Roberson, C.E., and Polzer, W.L., 1964, Sources of mineral constituents in water from granitic rocks, Sierra Nevada, California and Nevada: U.S. Geological Survey Water-Supply Paper 1535-I, 170 p.

Fontes, J.-C., and Garnier, J.-M., 1979, Determination of the initial ^{14}C activity of the total dissolved carbon: A review of the existing models and a new approach: Water Resources Research, v. 15, p. 399-413.

Foster, M.D., 1950, The origin of high sodium bicarbonate waters in the Atlantic and Gulf Coastal Plains: Geochimica et Cosmochimica Acta, v. 1, p. 33-48.

Friedman, I., and O'Neil, J.R., 1977, Compilation of stable isotope fractionation factors of geochemical interest, chap. KK *in* Fleischer, M., ed., Data of Geochemistry (6th ed.): U.S. Geological Survey Professional Paper 440, 12p.

Garrels, R.M., and Mackenzie, F.T., 1967, Origin of the chemical compositions of some springs and Lakes, *in* Equilibrium Concepts in Natural Water Systems: Advances in Chemistry Series, no. 67, American Chemical Society, Washington, D.C., p. 222-242.

Glynn, P.D., 1991, Effect of impurities in gypsum on contaminant transport at Pinal Creek, Arizona: U.S. Geological Survey Toxic Substances Hydrology Program--Proceedings of the Technical Meeting, Monterey California, March 11-15, 1991, Mallard, G.E., and Aronson, D.A., Eds., Water-Resources Investigations Report 91-4034, [in press].

Hahl, D.C., and Mitchell, C.G., 1963, Dissolved mineral inflow to Great Salt Lake and chemical characteristics of the salt lake brine. Part I: Selected hydrologic data: Water Resources Bulletin, v. 3, Pt. I, Salt Lake City, Utah, Utah Geological and Mineralogical Society, 40 p.

Hanshaw, B.B., Back, W., and Rubin, M., 1964, Radiocarbon determinations for estimating ground-water flow velocities in central Florida: Science, v. 148, no. 3669, p. 494-495.

Hoefs, J., 1973, Stable Isotope Geochemistry (3d ed.): New York, N.Y., Springer-Verlag, 241 p.

Hult, M.F., 1984, Ground-water contamination by crude oil at the Bemidjii, Minnesota research site: U.S. Geological Survey Toxic Waste--Ground-Water contamination study: U.S. Geological Survey Water-Resources Investigations Report 84-4188, 107 p.

Ingerson, E., and Pearson, F.J., Jr., 1964, Estimation of age and rate of motion of groundwater by the ^{14}C-method, *in* Recent Researches in the Fields of Atmosphere, Hydrosphere, and Nuclear Geochemistry, Sugawara Festival Volume: Maruzen Co., Tokyo, p. 263-283.

Lee, R.W., and Strickland, D.J., 1988, Geochemistry of groundwater in Tertiary and Cretaceous sediments of the southeastern coastal plain in eastern Georgia, South Carolina, and southeastern North Carolina: Water Resources Research, v. 24, p. 291-303.

Lovley, D.K., 1987, Organic matter mineralization with the reduction of ferric iron: A review: Geomicrobiology Journal, v. 5, p. 375-399, 1987.

Lovley, D.K., and Klug, M.J., 1986, Model for the distribution of sulfate reduction and methanogenesis in freshwater sediments: Geochimica et Cosmochimica Acta, v. 50, p. 11-18.

Mook, W.G., 1972, On the reconstruction of the initial ^{14}C content of groundwater from the chemical and isotopic composition, *in* Proceedings of Eighth International Conference on Radiocarbon Dating, v. 1: Royal Society of New Zealand, Wellington, New Zealand, p. 342-352.

Mook, W.G., 1976, The dissolution-exchange model for dating groundwater with ^{14}C, *in* Interpretation of Environmental Isotope and Hydrochemical Data in Groundwater Hydrology: International Atomic Energy Agency, Vienna, p. 213-225.

Mook, W.G., 1980, Carbon-14 in hydrogeological studies, *in* Fritz, P., and Fontes, J.Ch., eds., Handbook of Environmental Isotope Geochemistry, Volume 1, The Terrestrial Environment, A, Chapter 2: New York, N.Y., Elsevier Scientific Publishing Company, p. 49-74.

Mook, W.G., Bommerson, J.C., and Staverman, W.H., 1974, Carbon isotope fractionation between dissolved bicarbonate and gaseous carbon dioxide: Earth and Planetary Science Letters, v. 22, p. 169-176.

Ohmoto, H., and Rye, R.O., 1979, Isotopes of sulfur and carbon, *in* Barnes, A.L., ed., Geochemistry of Hydrothermal Ore Deposits: New York, Wiley-Interscience, p.509-567.

Parkhurst, D.L., Thorstenson, D.C., and Plummer, L.N., 1980, PHREEQE--A computer program for geochemical calculations: U.S. Geological Survey Water-Resources Investigation Report 80-96, 210 p.

Parkhurst, D.L., Plummer, L.N., and Thorstenson, D.C., 1982, BALANCE--A computer program for calculating mass transfer for geochemical reactions in ground water: U.S. Geological Survey Water-Resources Investigations Report 82-14, 29 p.

Pearson, F.J., Jr., and Rightmire, C.T., 1980, Sulphur and oxygen isotopes in aqueous sulphur compounds, *in* Fritz, P., and Fontes, J.Ch., eds., Handbook of Environmental Isotope Geochemistry, Volume 1, The Terrestrial Environment, A, Chapter 2: New York, N.Y., Elsevier Scientific Publishing Company, p. 227-258.

Plummer, L.N., 1984, Geochemical modeling: A comparison of forward and inverse methods, *in,* Hitchon, B., and Wallick, E.I., eds., First Canadian/American Conference on Hydrogeology, Practical Applications of Ground Water Geochemistry: Worthington, Ohio, National Water Well Association, p. 149-177.

Plummer, L.N., and Back, W., 1980, The mass balance approach: Applications to interpreting the chemical evolution of hydrologic systems: American Journal of Science, v. 280, p. 130-142.

Plummer, L.N., Jones, B.F., and Truesdell, A.H., 1976, WATEQF--A FORTRAN IV version of WATEQ, a computer program for calculating chemical equilibria of natural waters: U.S. Geological Survey Water-Resources Investigations Report 76-13, 61 p.

Plummer, L.N., Parkhurst, D.L., and Thorstenson, D.C., 1983, Development of reaction models for groundwater systems: Geochimica et Cosmochimica Acta, v. 47, p. 665-685.

Plummer, L.N., Parkhurst, D.L., Fleming, G.F., and Dunkle, S.A., 1988, A computer program incorporating Pitzer's equations for calculation of geochemical reactions in brines: U.S. Geological Survey Water-Resources Investigations Report 88-4153, 310 p.

Plummer, L.N., Busby, J.F., Lee, R.W., and Hanshaw, B.B., 1990, Geochemical modeling of the Madison aquifer in parts of Montana, Wyoming, and South Dakota: Water Resources Research, v. 26, p. 1981-2014.

Pucci, A.A., Jr., and Owens, J.P., 1989, Geochemical variations in a core of hydrogeologic units near Freehold, New Jersey: Ground Water, v. 27, p. 802-812.

Price, F.T., and Shieh, Y.N., 1979, Fractionation of sulfur isotopes during laboratory synthesis of pyrite at low temperatures: Chemical Geology, v. 27, p. 245-253.

Reardon, E.J., and Fritz, P., 1978, Computer modeling of groundwater ^{13}C and ^{14}C isotope compositions: Journal of Hydrology, v. 36, p. 201-224.

Reardon, E.J., Mozeto, A.A., and Fritz, P., 1980, Recharge in northern clime calcareous sandy soils: soil water chemical and carbon-14 evolution: Geochimica et Cosmochimica Acta, v. 44, p. 1723-1735.

Rye, R.O., Back, W., Hanshaw, B.B., Rightmire, C.T., and Pearson, F.J., Jr., 1981, The origin and isotopic composition of dissolved sulfide in groundwater from carbonate aquifers in Florida and Texas: Geochimica et Cosmochimica Acta, v. 45, p. 1941-1950.

Spencer, R.J., Eugster, H.P., Jones, B.F., and Rettig, S.L., 1985a, Geochemistry of Great Salt Lake, Utah I: Hydrochemistry since 1850: Geochimica et Cosmochimica Acta, v. 49, p. 727-737.

Spencer, R.J., Eugster, H.P., and Jones, B.F., 1985b, Geochemistry of Great Salt Lake, Utah II: Pleistocene-Holocene evolution: Geochimica et Cosmochimica Acta, v. 49, p. 739-747.

Stumm, W., and Morgan, J.J., 1981, Aquatic Chemistry, An Introduction Emphasizing Chemical Equilibria in Natural Waters: New York, Wiley, 780p.

Tamers, M.A., 1967, Surface-water infiltration and groundwater movement in arid zones of Venezuela, *in* Isotopes in Hydrology: International Atomic Energy Agency, Vienna, p. 339-351.

Tamers, M.A., 1975, Validity of radiocarbon dates on groundwater: Geophysical Surveys, v. 2, p. 217-239.

Tamers, M.A., and Scharpenseel, H.W., 1970, Sequential sampling of radiocarbon in groundwater, *in* Isotope Hydrology 1970: International Atomic Energy Agency, Vienna, p. 241-256.

Thode, H.G., Shima, M., Rees, C.E., and Krishnamurty, K.V., 1965, Carbon-13 isotope effects in systems containing carbon dioxide, bicarbonate, carbonate, and metal ions: Canadian Journal of Chemistry, v. 43, p. 582-595.

Thorstenson, D.C., Fisher, D.W., and Croft, M.G., 1979, The geochemistry of the Fox Hills-Basal Hell Creek aquifer in southwestern North Dakota and northwestern South Dakota: Water Resources Research, v. 15, p. 1479-1498.

Truesdell, A.H., and Jones, B.F., 1974, WATEQ, A computer program for calculating chemical equilibria of natural waters: Journal of Research, U.S. Geological Survey, v. 2, p. 233-274.

Vogel, J.C., 1967, Investigation of groundwater flow with radiocarbon: *in* Isotopes in Hydrology: International Atomic Energy Agency, Vienna, p. 255-368.

Vogel, J.C., and Ehhalt, D., 1963, The use of carbon isotopes in groundwater studies: *in* Radioisotopes in Hydrology: International Atomic Energy Agency, Vienna, p. 383-395.

Vogel, J.C., Grootes, P.M., and Mook, W.G., 1970, Isotope fractionation between gaseous and dissolved carbon dioxide: Zeitschrift für Physik, v. 230, p. 225-238.

Wigley, T.M.L., and Muller, A.B., 1981, Fractionation corrections in radiocarbon dating: Radiocarbon, v. 23, no. 2, p. 173-190.

Wigley, T.M.L., Plummer, L.N., and Pearson, F.J., Jr., 1978, Mass transfer and carbon isotope evolution in natural water systems: Geochimica et Cosmochimica Acta, v. 42, p. 1117-1139.

______ 1979, Errata: Geochimica et Cosmochimica Acta, v. 43, p. 1395.

ATTACHMENT A:

Listing of Source Code to DB; DB.FOR

```
      PROGRAM DBED
      CHARACTER*80 WLLNMS(50)
      CHARACTER*40 ADDRESS(50,5),LAT(50)
      CHARACTER*17 FORMATION(50)
      DIMENSION DBDATA(50,45)
      INTEGER DBSFG(50,45),NWLLS,TOTWELL,TOT(50)
      COMMON /DB/ DBDATA,DBSFG,NWLLS,WLLNMS,ADDRESS,LAT,FORMATION,IU(50,
     *4),IDEFAULT(5),TOTWELL,TOT
      CALL WELLFILE
      CALL MNLOOP
      END
      SUBROUTINE ADD (N)
      CHARACTER ANS
      CHARACTER*10 INPT,LABELS(45)
      CHARACTER*17 FORMATION(50)
      CHARACTER*80 LINE,WLLNMS(50)
      CHARACTER*40 ADDRESS(50,5),LAT(50),UPCS
      CHARACTER*9 ELEUNITS(8)
      CHARACTER*11 UNITS(0:4)
      CHARACTER*22 CARB(0:2)
      CHARACTER*13 PE(0:4)
      CHARACTER*12 DAVES(0:1)
      DIMENSION DBDATA(50,45), IUNITS(45), INEED(45)
      INTEGER DBSFG(50,45),ERR,NWLLS,TOTWELL,TOT(50),NONEG(45)
      COMMON /DB/ DBDATA,DBSFG,NWLLS,WLLNMS,ADDRESS,LAT,FORMATION,IU(50,
     *4),IDEFAULT(5),TOTWELL,TOT
      DATA ELEUNITS/'degrees C','mg/l     ','o/oo     ','% modern ','TU
     *      ','volts    ','feet     ','g/cm3    '/
      DATA IUNITS/1,0,2,-1,5,-1,-1,6,17*-1,2,0,8,3,4,4*3,2,0,-1,3*0,4*7,
     *0/
      DATA UNITS/'mmoles/l   ','meq/l      ','mg/l       ','ppm        '
     *,'mmol/kg H2O'/
      DATA CARB/'Uncorrected alkalinity','Corrected alkalinity  ','Total
     * carbon          '/
      DATA PE/'Redox ignored','Eh       ','Diss. O2 ','Sato Diss. O2','S
     *O4/H2S      '/
      DATA NONEG/0,0,1,1,0,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
     *1,0,1,0,0,0,0,1,1,1,0,0,0,0,1,1,1,0/
      DATA LABELS/'Temp.     ','pH        ','Diss. O2  ','Alkalinity','T
     *ritium   ','H2S       ','Ca2+      ','Eh        ','Mg2+      ','Na
     *+       ','K+        ','Cl-       ','SO42-     ','F-        ','SiO
     *2      ','Br-       ','B         ','Ba2+      ','Li+       ','Sr2+
     *      ','Fe        ','Mn        ','NO2+NO3-N ','NH4+kjd-N ','PO4-P
     *     ','DOC       ','Sp. Cond. ','Density   ','13C/12C   ','14C
     *    ','34SSO4    ','34SH2S    ','D         ','18O       ','Diss CH
     *4  ','Sr 87/86  ','Al3+      ',3*'          ','Distance  ','Depth
     *    ','Casing    ','Elevation ','          '/
      DATA DAVES/'Debye-Huckel','Davies      '/
      DATA INEED/26*1,0,10*1,3*2,4*0,2/
   10 FORMAT (A)
   20 CALL CLS
      IF (NWLLS.EQ.50) GO TO 500
      DO 30 I=NWLLS,N,-1
   30   CALL MOVE (I,I+1)
```

```
      NWLLS=NWLLS+1
      TOTWELL=TOTWELL+1
      TOT(N)=TOTWELL
   40 IF (IDEFAULT(1).EQ.1) WRITE (*,50)
      IF (IDEFAULT(1).EQ.2) WRITE (*,60)
   50 FORMAT (' Do you wish to input (1) All data, or (2) Data for ','NE
     *TPATH?',/,' (<Enter> for All data)')
   60 FORMAT (' Do you wish to input (1) All data, or (2) Data for',' NE
     *TPATH',/,' (<Enter> for NETPATH data)')
      READ (*,10) ANS
      IF (ANS.EQ.' ') GO TO 70
      READ (ANS,'(I1)') I
      IF (I.LT.1.OR.I.GT.2) GO TO 40
      IDEFAULT(1)=I
   70 WRITE (*,80) (I,UNITS(I),I=0,4),UNITS(IDEFAULT(2))
     *(:LENS(UNITS(IDEFAULT(2))))
   80 FORMAT (' Units choices:'/,5(I4,': ',A,/),/,' Input units for this
     * well (<Enter> to use ',A,')')
      READ (*,10) ANS
      IF (ANS.EQ.' ') GO TO 90
      READ (ANS,'(I1)',ERR=70) I
      IF (I.LT.0.OR.I.GT.4) GO TO 70
      IDEFAULT(2)=I
   90 IU(N,1)=IDEFAULT(2)
  100 WRITE (*,110) (I,DAVES(I),I=0,1),DAVES(IDEFAULT(5))
     *(:LENS(DAVES(IDEFAULT(5))))
  110 FORMAT (/' Activity coefficient calaculation choices:'/2(I4,': ',
     *A,/),/,' Input choice (<Enter> to use ',A,')')
      READ (*,10) ANS
      IF (ANS.EQ.' ') GO TO 120
      READ (ANS,'(I1)',ERR=100) I
      IF (I.LT.0.OR.I.GT.1) GO TO 100
      IDEFAULT(5)=I
  120 IU(N,4)=IDEFAULT(5)
      WLLNMS(N)=' '
      DO 130 I=1,5
  130   ADDRESS(N,I)=' '
      LAT(N)=' '
      FORMATION(N)=' '
      WRITE (*,140)
  140 FORMAT (/' Enter well name (max. 32 char.) :')
      READ (*,150) LINE
  150 FORMAT (A80)
      WLLNMS(N)(5:36)=LINE(1:32)
      IF (IDEFAULT(1).EQ.2) GO TO 230
      WRITE (*,160)
  160 FORMAT (/' Enter address, 1 line at a time (max. 40 char.) :'/)
      DO 170 I=1,5
        READ (*,150) LINE
  170   ADDRESS(N,I)=LINE(1:40)
      WRITE (*,180)
  180 FORMAT (/' Enter field id (max. 8 char.) :')
      READ (*,150) LINE
      WLLNMS(N)(38:45)=LINE(1:8)
      WRITE (*,190)
  190 FORMAT (/' Enter latitude.longitude :')
      READ (*,150) LINE
      LAT(N)=LINE(1:40)
      WRITE (*,200)
  200 FORMAT (/' Enter lat. long. id. no. (max. 15 char.) :')
      READ (*,150) LINE
```

```
    WLLNMS(N)(47:61)=LINE(1:15)                                        AB1050
    WRITE (*,210)                                                      AB1060
210 FORMAT (/' Enter date/time (YY/MM/DD @ HHMM) :')                   AB1070
    READ (*,150) LINE                                                  AB1080
    WLLNMS(N)(64:78)=LINE(1:15)                                        AB1090
    WRITE (*,220)                                                      AB1100
220 FORMAT (/' Enter aquifer (max. 17 char.) :')                       AB1110
    READ (*,150) LINE                                                  AB1120
    FORMATION(N)=LINE(1:17)                                            AB1130
230 DO 240 I=1,45                                                      AB1140
240   DBSFG(N,I)=-1                                                    AB1150
    DO 410 I=1,45                                                      AB1160
      IF (INEED(I).EQ.2) GO TO 410                                     AB1170
      IF (INEED(I).EQ.0.AND.IDEFAULT(1).EQ.2) GO TO 410                AB1180
      IF (I.NE.4) GO TO 300                                            AB1190
250   WRITE (*,260) (JJ,CARB(JJ),JJ=0,2),CARB(IDEFAULT(4))             AB1200
   *   (:LENS(CARB(IDEFAULT(4))))                                      AB1210
260   FORMAT (' Carbon choices:',/,3(I3,': ',A,/),/,' Input your choic AB1220
   *e (<Enter> to use ',A,')')                                         AB1230
      READ (*,10) ANS                                                  AB1240
      IF (ANS.EQ.' ') GO TO 270                                        AB1250
      READ (ANS,'(I1)',ERR=250) J                                      AB1260
      IF (J.LT.0.OR.J.GT.2) GO TO 250                                  AB1270
      IDEFAULT(4)=J                                                    AB1280
270   IU(N,3)=IDEFAULT(4)                                              AB1290
      WRITE (*,350) CARB(IU(N,3))(:LENS(CARB(IU(N,3)))),UNITS(IU(N,1)) AB1300
   *   (:LENS(UNITS(IU(N,1))))                                         AB1310
      READ (*,360,ERR=270) INPT                                        AB1320
      GO TO 370                                                        AB1330
280   WRITE (*,290)                                                    AB1340
290   FORMAT (/,' Alkalinity must be positive.'/' Enter total disolved AB1350
   *'' inorganic carbon.')                                             AB1360
      IDEFAULT(4)=2                                                    AB1370
      GO TO 270                                                        AB1380
300   IF (I.NE.8) GO TO 340                                            AB1390
310   WRITE (*,320) (IJ,PE(IJ),IJ=0,4),PE(IDEFAULT(3))                 AB1400
   *   (:LENS(PE(IDEFAULT(3))))                                        AB1410
320   FORMAT (' PE calc. choices:',/,5(I3,': ',A,/),/,' Input your cho AB1420
   *ice (<Enter> to use ',A,')')                                       AB1430
      READ (*,10) ANS                                                  AB1440
      IF (ANS.EQ.' ') GO TO 330                                        AB1450
      READ (ANS,'(I1)',ERR=310) J                                      AB1460
      IF (J.LT.0.OR.J.GT.4) GO TO 310                                  AB1470
      IDEFAULT(3)=J                                                    AB1480
330   IU(N,2)=IDEFAULT(3)                                              AB1490
      IF (IU(N,2).NE.1) GO TO 410                                      AB1500
340   IF (IUNITS(I).EQ.0) WRITE (*,350) LABELS(I)(:LENS(LABELS(I)))    AB1510
      IF (IUNITS(I).GT.0) WRITE (*,350) LABELS(I)(:LENS(LABELS(I))),   AB1520
   *   ELEUNITS(IUNITS(I))(:LENS(ELEUNITS(IUNITS(I))))                 AB1530
      IF (IUNITS(I).LT.0) WRITE (*,350) LABELS(I)(:LENS(LABELS(I))),   AB1540
   *   UNITS(IU(N,1))(:LENS(UNITS(IU(N,1))))                           AB1550
350   FORMAT (/' Enter value for ',A,:,' in ',A,'.')                   AB1560
      READ (*,360,ERR=340) INPT                                        AB1570
360   FORMAT (A10)                                                     AB1580
370   CALL GETNO (INPT,DBDATA(N,I),DBSFG(N,I),ERR)                     AB1590
      IF (ERR.GT.0) GO TO (440,460,480),ERR                            AB1600
      IF (I.EQ.4.AND.DBDATA(N,I).LE.0.AND.IU(N,3).NE.2) GO TO 280      AB1610
      IF (NONEG(I).EQ.1.AND.DBDATA(N,I).LT.0.) THEN                    AB1620
        IF (I.EQ.4) THEN                                               AB1630
          WRITE (*,380) CARB(IU(N,3))(:LENS(CARB(IU(N,3))))            AB1640
          GO TO 270                                                    AB1650
```

```
          ELSE                                                          AB1660
            WRITE (*,380) LABELS(I)(:LENS(LABELS(I)))                   AB1670
            GO TO 340                                                   AB1680
          END IF                                                        AB1690
        END IF                                                          AB1700
  380   FORMAT (/,1X,A,' must be non-negative. Please enter a new value. AB1710
     *'/)                                                               AB1720
        IF (I.NE.26) GO TO 410                                          AB1730
        IF (DBDATA(N,26).EQ.0..OR.DBSFG(N,26).LT.0.) GO TO 410          AB1740
        IF (DBSFG(N,45).EQ.-1) THEN                                     AB1750
          DBSFG(N,45)=0                                                 AB1760
          DBDATA(N,45)=0                                                AB1770
        END IF                                                          AB1780
  390   WRITE (*,400) DBDATA(N,45)                                      AB1790
  400   FORMAT (/' Enter average redox state of DOC (Enter for ',F6.2,') AB1800
     *')                                                                AB1810
        READ (*,360,ERR=390) INPT                                       AB1820
        IF (INPT.EQ.' ') GO TO 410                                      AB1830
        CALL GETNO (INPT,DBDATA(N,45),DBSFG(N,45),ERR)                  AB1840
        IF (ERR.GT.0) GO TO 390                                         AB1850
  410 CONTINUE                                                          AB1860
      CALL EDIT (N)                                                     AB1870
  420 WRITE (*,430)                                                     AB1880
  430 FORMAT (/' Do you want to add another well? (<Enter> for no)')    AB1890
      READ (*,'(a)',ERR=420) LINE                                       AB1900
      IF (UPCS(LINE).NE.'Y') RETURN                                     AB1910
      N=N+1                                                             AB1920
      GO TO 20                                                          AB1930
  440 WRITE (*,450)                                                     AB1940
  450 FORMAT (' ***Bad input, redo. ***')                               AB1950
      IF (I.EQ.4) GO TO 270                                             AB1960
      GO TO 340                                                         AB1970
  460 WRITE (*,470)                                                     AB1980
  470 FORMAT (' ***No decimal point in number, redo. ***')              AB1990
      IF (I.EQ.4) GO TO 270                                             AB2000
      GO TO 340                                                         AB2010
  480 WRITE (*,490)                                                     AB2020
  490 FORMAT (' ***Unknown error, try again. ***')                      AB2030
      IF (I.EQ.4) GO TO 270                                             AB2040
      GO TO 340                                                         AB2050
  500 WRITE (*,510)                                                     AB2060
  510 FORMAT (' No room to add more wells.'/' Hit return to continue.') AB2070
      READ (*,*) INPT                                                   AB2080
      RETURN                                                            AB2090
      END                                                               AB2100
      SUBROUTINE CHECK                                                  AC    10
      CHARACTER*80 WLLNMS(50)                                           AC    20
      CHARACTER*17 FORMATION(50)                                        AC    30
      CHARACTER*40 ADDRESS(50,5),LAT(50)                                AC    40
      DIMENSION DBDATA(50,45)                                           AC    50
      INTEGER DBSFG(50,45),NWLLS,TOTWELL,TOT(50)                        AC    60
      COMMON /DB/ DBDATA,DBSFG,NWLLS,WLLNMS,ADDRESS,LAT,FORMATION,IU(50, AC    70
     *4),IDEFAULT(5),TOTWELL,TOT                                        AC    80
      OPEN (6,FILE='CHECK')                                             AC    90
      DO 20 I=1,NWLLS                                                   AC   100
        IF (IU(I,1).EQ.2.OR.IU(I,1).EQ.3) THEN                          AC   110
          SUM1=DBDATA(I,7)/40.08*2                                      AC   120
          SUM1=SUM1+DBDATA(I,9)/24.305*2                                AC   130
          SUM1=SUM1+DBDATA(I,10)/22.98977                               AC   140
          SUM1=SUM1+DBDATA(I,11)/39.0983                                AC   150
          SUM1=SUM1+DBDATA(I,20)/87.62*2                                AC   160
```

```
          SUM1=SUM1+DBDATA(I,21)/55.847*2
          SUM1=SUM1+DBDATA(I,22)/54.938*2
          SUM1=SUM1+10.**(-DBDATA(I,2))*1000
          SUM1=SUM1+DBDATA(I,19)/6.941
          SUM1=SUM1+DBDATA(I,18)/137.33*2
          SUM1=SUM1+DBDATA(I,24)/18.0383
          SUM1=SUM1+DBDATA(I,37)/26.98154*3
          SUM2=DBDATA(I,4)/61.0171
C          IF (IU(I,3).EQ.2) SUM2=DBDATA(I,4)/12.011
          SUM2=SUM2+DBDATA(I,12)/35
          SUM2=SUM2+DBDATA(I,13)/96*2
          SUM2=SUM2+DBDATA(I,14)/19
          SUM2=SUM2+DBDATA(I,23)/14.0067
          SUM2=SUM2+DBDATA(I,16)/79.904
        ELSE IF (IU(I,1).EQ.0.OR.IU(I,1).EQ.4) THEN
          SUM1=DBDATA(I,7)*2
          SUM1=SUM1+DBDATA(I,9)*2
          SUM1=SUM1+DBDATA(I,10)
          SUM1=SUM1+DBDATA(I,11)
          SUM1=SUM1+DBDATA(I,20)*2
          SUM1=SUM1+DBDATA(I,21)*2
          SUM1=SUM1+DBDATA(I,22)*2
          SUM1=SUM1+10.**(-DBDATA(I,2))*1000
          SUM1=SUM1+DBDATA(I,19)
          SUM1=SUM1+DBDATA(I,18)*2
          SUM1=SUM1+DBDATA(I,24)
          SUM1=SUM1+DBDATA(I,37)*3
          SUM2=DBDATA(I,4)
          SUM2=SUM2+DBDATA(I,12)
          SUM2=SUM2+DBDATA(I,13)*2
          SUM2=SUM2+DBDATA(I,14)
          SUM2=SUM2+DBDATA(I,23)
          SUM2=SUM2+DBDATA(I,16)
        ELSE
          SUM1=DBDATA(I,7)
          SUM1=SUM1+DBDATA(I,9)
          SUM1=SUM1+DBDATA(I,10)
          SUM1=SUM1+DBDATA(I,11)
          SUM1=SUM1+DBDATA(I,20)
          SUM1=SUM1+DBDATA(I,21)
          SUM1=SUM1+DBDATA(I,22)
          SUM1=SUM1+10.**(-DBDATA(I,2))*1000
          SUM1=SUM1+DBDATA(I,19)
          SUM1=SUM1+DBDATA(I,18)
          SUM1=SUM1+DBDATA(I,24)
          SUM1=SUM1+DBDATA(I,37)
          SUM2=DBDATA(I,4)
          SUM2=SUM2+DBDATA(I,12)
          SUM2=SUM2+DBDATA(I,13)
          SUM2=SUM2+DBDATA(I,14)
          SUM2=SUM2+DBDATA(I,23)
          SUM2=SUM2+DBDATA(I,16)
        END IF
        PERCENT=(SUM1-SUM2)/(SUM1+SUM2)*100
        WRITE (6,10) WLLNMS(I)(5:36),PERCENT
   10   FORMAT (1X,A32,'         ',F8.3,'% error')
   20 CONTINUE
      CLOSE (6)
      RETURN
      END
      SUBROUTINE CLS
```

```
      WRITE (*,*) CHAR(27)//'[H'//CHAR(27)//'[J'                         AD  20
      RETURN                                                             AD  30
      END                                                                AD  40
      SUBROUTINE DELETE (N)                                              AE  10
      CHARACTER*80 WLLNMS(50)                                            AE  20
      CHARACTER*17 FORMATION(50)                                         AE  30
      CHARACTER*40 ADDRESS(50,5),LAT(50)                                 AE  40
      DIMENSION DBDATA(50,45)                                            AE  50
      INTEGER DBSFG(50,45),NWLLS,TOTWELL,TOT(50)                         AE  60
      COMMON /DB/ DBDATA,DBSFG,NWLLS,WLLNMS,ADDRESS,LAT,FORMATION,IU(50, AE  70
     *4),IDEFAULT(5),TOTWELL,TOT                                         AE  80
      DO 10 I=N+1,NWLLS                                                  AE  90
   10   CALL MOVE (I,I-1)                                                AE 100
      NWLLS=NWLLS-1                                                      AE 110
      RETURN                                                             AE 120
      END                                                                AE 130
      SUBROUTINE EDIT (N)                                                AF  10
      CHARACTER ANS                                                      AF  20
      CHARACTER*10 INPT,LABELS(45),PEVAL(0:4)                            AF  30
      CHARACTER*49 FRMT2                                                 AF  40
      CHARACTER*17 FORMATION(50)                                         AF  50
      CHARACTER*80 LINE,WLLNMS(50)                                       AF  60
      CHARACTER*40 ADDRESS(50,5),LAT(50)                                 AF  70
      CHARACTER*83 FRMT1                                                 AF  80
      CHARACTER*9 ELEUNITS(8)                                            AF  90
      CHARACTER*11 UNITS(0:4)                                            AF 100
      CHARACTER*22 CARB(0:2)                                             AF 110
      CHARACTER*13 PE(0:4)                                               AF 120
      CHARACTER*12 DAVES(0:1)                                            AF 130
      CHARACTER*32 FRMT3                                                 AF 140
      DIMENSION DBDATA(50,45), PRNT(5), IUNITS(45)                       AF 150
      INTEGER COUNT,DBSFG(50,45),ERR,NWLLS,TOTWELL,TOT(50)               AF 160
      COMMON /DB/ DBDATA,DBSFG,NWLLS,WLLNMS,ADDRESS,LAT,FORMATION,IU(50, AF 170
     *4),IDEFAULT(5),TOTWELL,TOT                                         AF 180
      DATA ELEUNITS/'degrees C','mg/l     ','o/oo     ','% modern ','TU  AF 190
     *      ','volts    ','feet     ','g/cm3    '/                       AF 200
      DATA IUNITS/1,0,2,-1,5,-1,-1,6,17*-1,2,0,8,3,4,4*3,2,3,-1,3*0,4*7, AF 210
     *0/                                                                 AF 220
      DATA UNITS/'mmoles/l   ','meq/l      ','mg/l       ','ppm        ' AF 230
     *,'mmol/kg H2O'/                                                    AF 240
      DATA CARB/'Uncorrected alkalinity','Corrected alkalinity  ','Total AF 250
     * carbon          '/                                                AF 260
      DATA PE/'Redox ignored','Eh       ','Diss. O2 ','Sato Diss. O2','S AF 270
     *O4/H2S      '/                                                     AF 280
      DATA LABELS/'Temp.     ','pH        ','diss. O2  ','Alkalinity','T AF 290
     *ritium   ','H2S       ','Ca2+      ','Eh        ','Mg2+      ','Na AF 300
     *+       ','K+        ','Cl-       ','SO42-     ','F-        ','SiO AF 310
     *2      ','Br-       ','B         ','Ba2+      ','Li+       ','Sr2+ AF 320
     *      ','Fe        ','Mn        ','NO2+NO3-N ','NH4+kjd-N ','PO4-P AF 330
     *     ','DOC       ','Sp. Cond. ','Density   ','13C/12C   ','14C    AF 340
     *    ','34SSO4    ','34SH2S    ','D         ','18O       ','Diss CH AF 350
     *4  ','Sr 87/86  ','Al3+      ',3*'          ','Distance  ','Depth  AF 360
     *    ','Casing    ','Elevation ','          '/                      AF 370
      DATA PEVAL/'Redox Ign.','NOT USED  ','Diss. O2  ','Sato O2   ','SO AF 380
     *4/H2S   '/                                                         AF 390
      DATA FRMT1/'(/ 4x,************,4x,************,4x,************,4x, AF 400
     *************,4x,************)'/                                    AF 410
      DATA FRMT2/'(//'' The current value of '',a10,'' is           )'/ AF 420
   10 FORMAT (A)                                                         AF 430
   20 CALL CLS                                                           AF 440
      WRITE (*,30) WLLNMS(N)(5:79)                                       AF 450
```

```
   30 FORMAT (' 41) ',A75,/)
      IF (IU(N,3).EQ.0) LABELS(4)='Alkalinity'
      IF (IU(N,3).EQ.1) LABELS(4)='Corr. Alk.'
      IF (IU(N,3).EQ.2) LABELS(4)='Total C   '
      WRITE (*,40) CHAR(27)//'[H',(I,LABELS(I),I=1,37)
   40 FORMAT (1X,A4,/,8(//3X,5(I3,')',A10)))
      WRITE (*,'(1X,a4//)') CHAR(27)//'[H'
      DO 110 I=1,36,5
        COUNT=0
        DO 100 J=I,I+4
          IF (J.NE.8) GO TO 50
C SPECIAL CASE: Eh
          IF (IU(N,2).EQ.1) GO TO 50
          FRMT1((J-I)*16+7:(J-I)*16+18)=''''//PEVAL(IU(N,2))//''''
          GO TO 100
   50     IF (LABELS(J).EQ.' ') GO TO 70
          IF (DBSFG(N,J).LT.0) GO TO 80
          FRMT1((J-I)*16+7:(J-I)*16+18)='         f10.'//CHAR(DBSFG(N,J)+
     *     48)
   60     COUNT=COUNT+1
          PRNT(COUNT)=DBDATA(N,J)
          GO TO 100
   70     FRMT1((J-I)*16+7:(J-I)*16+18)='''          '''
          GO TO 100
   80     IF (DBSFG(N,J).EQ.-2) GO TO 90
          FRMT1((J-I)*16+7:(J-I)*16+18)='''**********'''
          GO TO 100
   90     FRMT1((J-I)*16+7:(J-I)*16+18)='''<'',f9.3  '
          GO TO 60
  100   CONTINUE
  110   WRITE (*,FRMT1) (PRNT(J),J=1,COUNT)
      WRITE (*,120)
  120 FORMAT (' Enter # of value to change, 0 to exit :')
      READ (*,*,ERR=20) I
      IF (I.EQ.0) RETURN
      IF (I.GT.41.OR.I.LT.0) GO TO 20
      IF (I.EQ.41) GO TO 310
      IF (LABELS(I).EQ.' ') GO TO 20
      IF (I.NE.4) GO TO 180
  130 WRITE (*,140) (JJ,CARB(JJ),JJ=0,2),CARB(IU(N,3))(:LENS(CARB(IU(N,
     *3))))
  140 FORMAT (' Carbon choices:',/,3(I3,': ',A,/),/,' Input your choice
     *(<Enter> to use ',A,')')
      READ (*,10) ANS
      IF (ANS.EQ.' ') GO TO 150
      READ (ANS,'(I1)',ERR=130) J
      IF (J.LT.0.OR.J.GT.2) GO TO 130
      IU(N,3)=J
  150 IF (IU(N,3).EQ.0) LABELS(4)='Alkalinity'
      IF (IU(N,3).EQ.1) LABELS(4)='Corr. Alk.'
      IF (IU(N,3).EQ.2) LABELS(4)='Total C   '
      GO TO 220
  160 IU(N,3)=2
      WRITE (*,170)
  170 FORMAT (/,' Alkalinity cannot be negative.',/)
      GO TO 130
  180 IF (I.NE.8) GO TO 220
  190 WRITE (*,200) (IJ,PE(IJ),IJ=0,4),PE(IU(N,2))(:LENS(PE(IU(N,2))))
  200 FORMAT (' PE calc. choices:',/,5(I3,': ',A,/),/,' Input your choic
     *e (<Enter> to use ',A,')')
      READ (*,10) ANS
```

```
      IF (ANS.EQ.' ') GO TO 210
      READ (ANS,'(I1)',ERR=190) J
      IF (J.LT.0.OR.J.GT.4) GO TO 190
      IU(N,2)=J
  210 IF (IU(N,2).NE.1) GO TO 20
  220 IF (DBSFG(N,I).GE.0) GO TO 240
      IF (DBSFG(N,I).EQ.-2) GO TO 230
      FRMT2(38:48)='undefined.'''
      WRITE (*,FRMT2) LABELS(I)
      GO TO 260
  230 FRMT2(38:48)=''',''<'',f9.3 '
      GO TO 250
  240 FRMT2(38:48)=''',f10.'//CHAR(DBSFG(N,I)+48)//'    '
  250 WRITE (*,FRMT2) LABELS(I),DBDATA(N,I)
  260 IF (IUNITS(I).EQ.0) WRITE (*,270)
      IF (IUNITS(I).GT.0) WRITE (*,280) ELEUNITS(IUNITS(I))
     *(:LENS(ELEUNITS(IUNITS(I))))
      IF (IUNITS(I).EQ.-1) WRITE (*,280) UNITS(IU(N,1))
     *(:LENS(UNITS(IU(N,1))))
  270 FORMAT (/' Enter new value :')
  280 FORMAT (/' Enter new value (in ',A,'):')
      READ (*,'(a10)') INPT
      CALL GETNO (INPT,DBDATA(N,I),DBSFG(N,I),ERR)
      IF (ERR.GT.0) GO TO (390,410,430),ERR
      IF (I.EQ.4.AND.DBDATA(N,I).LT.0.AND.IU(N,3).LT.2) GO TO 160
      IF (I.NE.26) GO TO 20
  290 WRITE (*,300) DBDATA(N,45)
  300 FORMAT (/' Enter average redox state of DOC (Enter for ',F6.2,')')
      READ (*,'(A10)') INPT
      IF (INPT.EQ.' ') GO TO 20
      CALL GETNO (INPT,DBDATA(N,45),DBSFG(N,45),ERR)
      IF (ERR.GT.0) GO TO 290
      GO TO 20
  310 CALL CLS
      WRITE (*,320) WLLNMS(N)(5:36),WLLNMS(N)(38:45),WLLNMS(N)(47:61),
     *WLLNMS(N)(64:78),LAT(N),ADDRESS(N,1),ADDRESS(N,2),ADDRESS(N,3),
     *ADDRESS(N,4),ADDRESS(N,5),FORMATION(N)
  320 FORMAT (//,' The current well   1)  name     = ',A32/'
     *        2) field id.  = ',A8/'                     3)  id no.    =
     *',A15/'                     4) date/time  = ',A15/'
     *    5) lat/long   = ',A40/'                     6) address 1  = ',
     *A40/'                     7) address 2  = ',A40/'
     *  8) address 3  = ',A40/'                     9) address 4  = ',
     *A40/'                    10) address 5  = ',A40/'
     * 11) aquifer    = ',A17)
      DO 340 K=1,4
        IF (DBSFG(N,K+40).GE.0) THEN
          FRMT3='(19X,I2,'') '',A10,'' = '',F10.X)'
          WRITE (FRMT3,330) FRMT3(1:27),DBSFG(N,K+40),FRMT3(29:)
          WRITE (*,FRMT3) K+11,LABELS(K+40),DBDATA(N,K+40)
        ELSE
          FRMT3='(19X,I2,'') '',A10,'' = undefined'')'
          WRITE (*,FRMT3) K+11,LABELS(K+40)
        END IF
  330   FORMAT (A,I1,A)
  340 CONTINUE
      WRITE (*,350)
  350 FORMAT ('  Enter # to change, 0 to exit :')
      READ (*,*,ERR=310) J
      IF (J) 310,20,360
  360 IF (J.GT.15) GO TO 310
```

```
      IF (J.GT.11) GO TO 380                                            AF1680
      WRITE (*,370)                                                     AF1690
  370 FORMAT (/' Enter new value :')                                    AF1700
      READ (*,'(a80)') LINE                                             AF1710
      IF (J.EQ.1) WLLNMS(N)(5:36)=LINE(1:32)                            AF1720
      IF (J.EQ.2) WLLNMS(N)(38:45)=LINE(1:8)                            AF1730
      IF (J.EQ.3) WLLNMS(N)(47:61)=LINE(1:15)                           AF1740
      IF (J.EQ.4) WLLNMS(N)(64:78)=LINE(1:15)                           AF1750
      IF (J.EQ.5) LAT(N)=LINE(1:40)                                     AF1760
      IF (J.GT.5.AND.J.LT.11) ADDRESS(N,J-5)=LINE(1:40)                 AF1770
      IF (J.EQ.11) FORMATION(N)=LINE(1:17)                              AF1780
      GO TO 310                                                         AF1790
  380 WRITE (*,370)                                                     AF1800
      READ (*,'(a80)') LINE                                             AF1810
      CALL GETNO (LINE(1:10),DBDATA(N,J+29),DBSFG(N,J+29),ERR)          AF1820
      IF (ERR.GT.0) GO TO 380                                           AF1830
      GO TO 310                                                         AF1840
  390 WRITE (*,400)                                                     AF1850
  400 FORMAT (' ***Bad input, redo. ***')                               AF1860
      GO TO 260                                                         AF1870
  410 WRITE (*,420)                                                     AF1880
  420 FORMAT (' ***No decimal point in number, redo. ***')              AF1890
      GO TO 260                                                         AF1900
  430 WRITE (*,440)                                                     AF1910
  440 FORMAT (' ***Unknown error, try again. ***')                      AF1920
      GO TO 260                                                         AF1930
      END                                                               AF1940
      SUBROUTINE GETNO (INPT,RESULT,SIGFIG,ERROR)                       AG  10
      CHARACTER*10 INPT                                                 AG  20
      INTEGER ERROR,SIGFIG                                              AG  30
      REAL RESULT                                                       AG  40
      ERROR=0                                                           AG  50
      IF (INPT.EQ.'          ') GO TO 10                                AG  60
      IF (INPT(1:3).EQ.'***') GO TO 10                                  AG  70
      IF (INPT(1:1).EQ.'<') GO TO 20                                    AG  80
      GO TO 30                                                          AG  90
   10 RESULT=0.0                                                        AG 100
      SIGFIG=-1                                                         AG 110
      RETURN                                                            AG 120
   20 READ (INPT(2:10),'(F9.0)',ERR=90) RESULT                          AG 130
      SIGFIG=-2                                                         AG 140
      RETURN                                                            AG 150
   30 READ (INPT,'(F10.0)',ERR=90) A                                    AG 160
      DO 40 I=1,10                                                      AG 170
        IF (INPT(I:I).EQ.'.') GO TO 50                                  AG 180
   40 CONTINUE                                                          AG 190
C                                                                       AG 200
C Remove the following comment to force the use of decimals            AG 210
C      GO TO 600                                                        AG 220
      J=I                                                               AG 230
      GO TO 70                                                          AG 240
   50 DO 60 J=10,I,-1                                                   AG 250
        IF (INPT(J:J).NE.' ') GO TO 70                                  AG 260
   60 CONTINUE                                                          AG 270
      GO TO 80                                                          AG 280
   70 RESULT=A                                                          AG 290
      SIGFIG=J-I                                                        AG 300
      RETURN                                                            AG 310
   80 ERROR=ERROR+1                                                     AG 320
      ERROR=ERROR+1                                                     AG 330
   90 ERROR=ERROR+1                                                     AG 340
```

```
      SIGFIG=-1                                                         AG 350
      RESULT=0.0                                                        AG 360
      RETURN                                                            AG 370
      END                                                               AG 380
      FUNCTION LENS (STRING)                                            AH  10
      CHARACTER*(*) STRING                                              AH  20
      K=LEN(STRING)                                                     AH  30
      DO 10 I=K,1,-1                                                    AH  40
        IF (STRING(I:I).NE.' ') GO TO 20                                AH  50
   10 CONTINUE                                                          AH  60
      LENS=0                                                            AH  70
      RETURN                                                            AH  80
   20 LENS=I                                                            AH  90
      RETURN                                                            AH 100
      END                                                               AH 110
      SUBROUTINE MNLOOP                                                 AI  10
      CHARACTER*80 WLLNMS(50),LINE                                      AI  20
      CHARACTER*17 FORMATION(50)                                        AI  30
      CHARACTER*40 ADDRESS(50,5),LAT(50),UPCS                           AI  40
      CHARACTER ANS                                                     AI  50
      DIMENSION DBDATA(50,45)                                           AI  60
      INTEGER COUNT,DBSFG(50,45),NWLLS,TOTWELL,TOT(50)                  AI  70
      COMMON /DB/ DBDATA,DBSFG,NWLLS,WLLNMS,ADDRESS,LAT,FORMATION,IU(50,AI  80
     *4),IDEFAULT(5),TOTWELL,TOT                                        AI  90
      CALL RDDB                                                         AI 100
   10 COUNT=1                                                           AI 110
   20 CALL CLS                                                          AI 120
      WRITE (*,30)                                                      AI 130
   30 FORMAT ('  #  Well Name',26X,'| #  Well Name'/,1X,39('='),'|',39('AI 140
     *='))                                                              AI 150
      DO 40 I=1,15                                                      AI 160
        LINE='                                       |'                 AI 170
        IF (COUNT.GT.NWLLS) GO TO 90                                    AI 180
        WRITE (LINE,50) COUNT,WLLNMS(COUNT)(5:36)                       AI 190
        COUNT=COUNT+1                                                   AI 200
        IF (COUNT.GT.NWLLS) GO TO 90                                    AI 210
        WRITE (LINE,60) LINE(1:40),COUNT,WLLNMS(COUNT)(5:36)            AI 220
        WRITE (*,70) LINE(1:LENS(LINE))                                 AI 230
        COUNT=COUNT+1                                                   AI 240
   40 CONTINUE                                                          AI 250
   50 FORMAT (I2,') ',A32,3X,'|')                                       AI 260
   60 FORMAT (A40,I2,') ',A32)                                          AI 270
   70 FORMAT (1X,A)                                                     AI 280
   80 FORMAT (A)                                                        AI 290
      GO TO 110                                                         AI 300
   90 DO 100 J=15,I,-1                                                  AI 310
        WRITE (*,70) LINE(1:LENS(LINE))                                 AI 320
  100   LINE='                                       |'                 AI 330
      COUNT=1                                                           AI 340
  110 WRITE (*,120)                                                     AI 350
  120 FORMAT (1X,79('='))                                               AI 360
  130 IF (NWLLS.GT.30) WRITE (*,140)                                    AI 370
      IF (NWLLS.LE.30) WRITE (*,150)                                    AI 380
  140 FORMAT (' <A>dd, <D>elete, <E>dit, <M>ove, <N>ext page, ','<P>rintAI 390
     *, <S>ave, or <Q>uit?')                                            AI 400
  150 FORMAT (' <A>dd, <D>elete, <E>dit, <M>ove, ','<P>rint, <S>ave, or AI 410
     *<Q>uit?')                                                         AI 420
      READ (*,80,ERR=160,END=160) ANS                                   AI 430
      IF (UPCS(ANS).EQ.'N'.AND.NWLLS.GT.30) GO TO 20                    AI 440
      IF (UPCS(ANS).EQ.'E') GO TO 170                                   AI 450
```

```
      IF (UPCS(ANS).EQ.'A') GO TO 190
      IF (UPCS(ANS).EQ.'D') GO TO 210
      IF (UPCS(ANS).EQ.'M') GO TO 230
      IF (UPCS(ANS).EQ.'P') GO TO 340
      IF (UPCS(ANS).EQ.'S') GO TO 330
      IF (UPCS(ANS).EQ.'Q') GO TO 290
  160 CALL POSCUR
      GO TO 130
  170 CALL POSCUR
      WRITE (*,180)
  180 FORMAT (' Enter well # to edit (<Enter> to abort): ')
      READ (*,310,ERR=170,END=170) I
      IF (I.EQ.0) GO TO 160
      IF (I.GT.NWLLS) GO TO 170
      CALL EDIT (I)
      GO TO 10
  190 CALL POSCUR
      WRITE (*,200)
  200 FORMAT (' Add new well(s) at well # (<Enter> for ','end of list, 0
     * to abort):')
      READ (*,80,ERR=190,END=190) LINE
      IF (LINE.EQ.' ') THEN
        CALL ADD (NWLLS+1)
        GO TO 10
      END IF
      READ (LINE,310) I
      IF (I.EQ.0) GO TO 160
      IF (I.GT.NWLLS+1) GO TO 190
      CALL ADD (I)
      GO TO 10
  210 CALL POSCUR
      WRITE (*,220)
  220 FORMAT (' Enter well # to delete (<Enter> to abort)')
      READ (*,80,ERR=190,END=190) LINE
      READ (LINE,310) I
      IF (I.EQ.0) GO TO 160
      IF (I.GT.NWLLS) GO TO 210
      CALL DELETE (I)
      GO TO 10
  230 CALL POSCUR
      IF (NWLLS.LT.50) GO TO 250
      WRITE (*,240)
  240 FORMAT (' Can not move with 50 wells in data base. Hit <Enter>','
     *to continue.')
      READ (*,'(a80)') LINE
      GO TO 160
  250 WRITE (*,260)
  260 FORMAT (' Enter well # to move (<Enter> to abort)')
      READ (*,310,ERR=230,END=230) I
      IF (I.EQ.0) GO TO 160
      IF (I.GT.NWLLS) GO TO 230
  270 CALL POSCUR
      WRITE (*,280) I
  280 FORMAT (' Move well',I3,') to well number (<Enter> to abort) :')
      READ (*,310,ERR=270,END=270) J
      IF (J.EQ.0) GO TO 230
      IF (J.GT.NWLLS) GO TO 270
      CALL SWITCH (I,J,*160)
      GO TO 10
  290 CALL POSCUR
      WRITE (*,300)
```

```
  300 FORMAT (' Do you really want to quit (<Enter> quits)?')
      READ (*,80,ERR=290,END=290) ANS
  310 FORMAT (I5)
      IF (UPCS(ANS).NE.'N') THEN
        CALL CLS
        CALL WATEQFP
        WRITE (*,320)
        RETURN
      END IF
      GO TO 160
  320 FORMAT (' DB finished - Thank You.'/)
  330 CALL SAVE
      GO TO 10
  340 CALL POSCUR
      WRITE (*,350)
  350 FORMAT (' Enter well # to print (0 to quit, 51 for all,',' 99 to c
     *heck data) : ')
      READ (*,310,ERR=340,END=340) I
      JJ=0
      IF (I.EQ.0) GO TO 160
      IF (I.EQ.99) GO TO 370
      IF (I.LE.NWLLS) GO TO 380
      IF (I.NE.51) GO TO 340
      DO 360 JJ=1,NWLLS
        CALL PUTDATA (JJ)
  360 CONTINUE
      GO TO 10
  370 CALL CHECK
      GO TO 10
  380 CALL PUTDATA (I)
      GO TO 10
      END
      SUBROUTINE MOVE (S,D)
      CHARACTER*80 WLLNMS(50)
      CHARACTER*17 FORMATION(50)
      CHARACTER*40 ADDRESS(50,5),LAT(50)
      DIMENSION DBDATA(50,45)
      INTEGER D,DBSFG(50,45),NWLLS,S,TOTWELL,TOT(50)
      COMMON /DB/ DBDATA,DBSFG,NWLLS,WLLNMS,ADDRESS,LAT,FORMATION,IU(50,
     *4),IDEFAULT(5),TOTWELL,TOT
      WLLNMS(D)=WLLNMS(S)
      TOT(D)=TOT(S)
      LAT(D)=LAT(S)
      FORMATION(D)=FORMATION(S)
      DO 10 I=1,5
        ADDRESS(D,I)=ADDRESS(S,I)
   10 CONTINUE
      DO 20 I=1,45
        DBDATA(D,I)=DBDATA(S,I)
   20   DBSFG(D,I)=DBSFG(S,I)
      DO 30 I=1,4
   30   IU(D,I)=IU(S,I)
      RETURN
      END
      SUBROUTINE PUTDATA (I)
      CHARACTER*80 WLLNMS(50),LINE
      CHARACTER*40 ADDRESS(50,5),LAT(50)
      CHARACTER*17 FORMATION(50)
      CHARACTER*16 WORDS(45),CARB(0:2),TEMPWD
      CHARACTER*11 UNITS(0:4),TEMPUN
      CHARACTER*10 MAT
```

```
      CHARACTER*9 ELEUNITS(0:8)                                          AK  80
      CHARACTER*8 FNAME,TEMPVL                                           AK  90
      CHARACTER*7 REPORT(45)                                             AK 100
      DIMENSION DBDATA(50,45), IUNITS(45)                                AK 110
      INTEGER DBSFG(50,45),NWLLS,NUM(45),TOTWELL,TOT(50)                 AK 120
      COMMON /DB/ DBDATA,DBSFG,NWLLS,WLLNMS,ADDRESS,LAT,FORMATION,IU(50, AK 130
     *4),IDEFAULT(5),TOTWELL,TOT                                         AK 140
      DATA NUM/1,14,27,16,2,23,8,24,28,25,3,17,4,37,7,15,9,26,10,45,11,  AK 150
     *35,18,29,20,30,21,31,22,32,19,36,12,33,13,34,6,5,7*0/              AK 160
      DATA WORDS/                                                        AK 170
     * 'Temperature     ','pH              '                             AK 180
     *,'Dissolved Oxygen','Alkalinity      '                             AK 190
     *,'Tritium         ','Hydrogen Sulfide'                             AK 200
     *,'Calcium         ','Eh              '                             AK 210
     *,'Magnesium       ','Sodium          '                             AK 220
     *,'Potassium       ','Chloride        '                             AK 230
     *,'Sulfate         ','Fluoride        '                             AK 240
     *,'Silica          ','Bromide         '                             AK 250
     *,'Boron           ','Barium          '                             AK 260
     *,'Lithium         ','Strontium       '                             AK 270
     *,'Iron            ','Manganese       '                             AK 280
     *,'Nitrite-Nitrate ','Nh4 (Kjd)       '                             AK 290
     *,'Phosphate       ','DOC             '                             AK 300
     *,'Sp. Cond.       ','Density         '                             AK 310
     *,'Delta C-13 TDIC ','Carbon 14 TDIC  '                             AK 320
     *,'Delta S-34 (SO4)','Delta S-34 (H2S)'                             AK 330
     *,'Delta Deuterium ','Delta O-18      '                             AK 340
     *,'CH4 (aq)        ','Sr 87/86        '                             AK 350
     *,'Aluminum        ','                '                             AK 360
     *,'                ','                '                             AK 370
     *,'Distance        ','Depth           '                             AK 380
     *,'Casing          ','Elevation       '                             AK 390
     *,'RS of DOC       '/                                               AK 400
      DATA REPORT/14*'       ','as SiO2',7*'       ',2*'as N   ','as P   AK 410
     *   ','as C   ',19*'       '/                                       AK 420
      DATA ELEUNITS/'         ','degrees C','mg/l     ','o/oo     ','% m AK 430
     *odern ','TU       ','volts    ','feet     ','g/cm3    '/           AK 440
      DATA IUNITS/1,0,2,-1,5,-1,-1,6,17*-1,2,0,8,3,4,4*3,2,0,-1,3*0,4*7, AK 450
     *0/                                                                 AK 460
      DATA UNITS/'mmoles/l   ','meq/l      ','mg/l       ','ppm        ' AK 470
     *,'mmol/kg H2O'/                                                    AK 480
      DATA CARB/'Alkalinity #    ','Alkalinity #    ','Total carbon #  ' AK 490
     */                                                                  AK 500
      FNAME='output'                                                     AK 510
      IC=ICHAR('0')                                                      AK 520
      FNAME(7:8)=CHAR(INT(I/10)+IC)//CHAR(MOD(I,10)+IC)                  AK 530
      OPEN (7,FILE=FNAME)                                                AK 540
      WRITE (7,10) WLLNMS(I)(5:36)                                       AK 550
   10 FORMAT (/' Well name          : ',A/)                              AK 560
      WRITE (7,30) ADDRESS(I,1)                                          AK 570
      DO 20 J=2,5                                                        AK 580
        WRITE (7,40) ADDRESS(I,J)                                        AK 590
   20 CONTINUE                                                           AK 600
   30 FORMAT (' Owner              : ',A)                                AK 610
   40 FORMAT ('                      ',A)                                AK 620
      WRITE (7,50) WLLNMS(I)(47:61)                                      AK 630
   50 FORMAT (' Site ID            : ',A)                                AK 640
      WRITE (7,60) LAT(I)                                                AK 650
   60 FORMAT (' Latitude/longitude : ',A)                                AK 660
      WRITE (7,70) WLLNMS(I)(64:78)                                      AK 670
   70 FORMAT (' Date/time sampled  : ',A/)                               AK 680
```

```
      DO 140 J=1,38,2
        LINE=' '
        DO 130 K=0,1
          IF (NUM(J+K).EQ.0) GO TO 130
   80     FORMAT (A)
          TEMPWD=WORDS(NUM(J+K))
          IF (NUM(J+K).EQ.4) TEMPWD=CARB(IU(I,3))
          LINE(K*38+1:K*38+16)=TEMPWD
          IF (NUM(J+K).EQ.8.AND.IU(I,2).NE.1) GO TO 100
          IF (DBSFG(I,NUM(J+K)).LT.0) GO TO 90
          MAT='(F8.'//CHAR(DBSFG(I,NUM(J+K))+48)//')'
          WRITE (TEMPVL,MAT) DBDATA(I,NUM(J+K))
          LINE(18+38*K:25+38*K)=TEMPVL
          GO TO 120
   90     IF (DBSFG(I,NUM(J+K)).EQ.-2) GO TO 110
  100     LINE(18+38*K:25+38*K)='    N.D.'
          GO TO 120
  110     MAT='(''<'',F7.3)'
          WRITE (TEMPVL,MAT) DBDATA(I,NUM(J+K))
          LINE(18+38*K:25+38*K)=TEMPVL
  120     TEMPUN=UNITS(IU(I,1))
          IF (IUNITS(NUM(J+K)).NE.-1) TEMPUN=ELEUNITS(IUNITS(NUM(J+K)))
          LINE(27+38*K:37+38*K)=TEMPUN
          IF (K.EQ.0) GO TO 130
          IF (IUNITS(NUM(J+K)).NE.-1) GO TO 130
          IOFF=LENS(UNITS(IU(I,1)))
          LINE(66+IOFF:)=REPORT(NUM(J+K))
  130   CONTINUE
        IF (LINE.NE.' ') WRITE (7,80) LINE
  140 CONTINUE
      WRITE (7,80) '--------------------------------------'
      WRITE (7,80) 'N.D. = not determined'
      WRITE (7,80) 'TDIC = Total Dissolved Inorganic Carbon'
      WRITE (7,80) 'DOC = Dissolved Organic Carbon'
      WRITE (7,80) 'Sp. Cond. = Specific Conductivity (uS/cm)'
      WRITE (7,80) 'RS = Redox State'
      IF (IU(I,3).EQ.0) THEN
        WRITE (7,80) '# = uncorrected, reported as HCO3-'
      ELSE IF (IU(I,3).EQ.1) THEN
        WRITE (7,80) '# = corrected, reported as HCO3-'
      ELSE
        WRITE (7,80) '# = total carbon, reported as HCO3-'
      END IF
      CLOSE (7)
      RETURN
      END
      SUBROUTINE POSCUR
      WRITE (*,10) CHAR(27)//'[H'
   10 FORMAT (1X,A,//////////////////)
      WRITE (*,*) CHAR(27)//'[B'//CHAR(27)//'[B'//CHAR(27)//'[K'//
     *CHAR(27)//'[A'//CHAR(27)//'[K'//CHAR(27)//'[A'//CHAR(27)//'[K'//
     *CHAR(27)//'[A'
      RETURN
      END
      SUBROUTINE RDDB
      CHARACTER*1 UTS
      CHARACTER*8 UNITS(0:4)
      CHARACTER*10 INPT
      CHARACTER*80 LINE,WLLNMS(50)
      CHARACTER*40 ADDRESS(50,5),LAT(50),DFILE
      CHARACTER*17 FORMATION(50)
```

```
      CHARACTER*3 ECP,TEMP,EP2
      DIMENSION DBDATA(50,45)
      LOGICAL NEW,NEW2
      INTEGER DBSFG(50,45),ERR,NWLLS,ERIC,JCOUNTER,TOTWELL,TOT(50)
      COMMON /DB/ DBDATA,DBSFG,NWLLS,WLLNMS,ADDRESS,LAT,FORMATION,IU(50,
     *4),IDEFAULT(5),TOTWELL,TOT
      COMMON /FILE/ DFILE,ISAVE
      IDEFAULT(1)=1
      IDEFAULT(2)=2
      IDEFAULT(3)=0
      IDEFAULT(4)=0
      IDEFAULT(5)=0
      NWLLS=0
      TOTWELL=0
      OPEN (UNIT=5,FILE=DFILE//'.LON',STATUS='OLD',ERR=90)
   10 READ (5,40,ERR=90,END=90) LINE
      NWLLS=NWLLS+1
      IF (NWLLS.EQ.51) GO TO 180
      READ (LINE,30,ERR=120) (IU(NWLLS,JK),JK=1,4)
      DO 20 ID=1,4
   20   IDEFAULT(ID+1)=IU(NWLLS,ID)
   30 FORMAT (4(I1))
   40 FORMAT (A80)
      WLLNMS(NWLLS)=LINE
      READ (5,40,END=100) LINE
      LAT(NWLLS)=LINE(1:40)
      READ (LINE(66:70),50) TOT(NWLLS)
      IF (NWLLS.EQ.1) READ (LINE(75:79),50) TOTWELL
   50 FORMAT (I5)
      IF (TOT(NWLLS).EQ.0) THEN
        TOT(NWLLS)=NWLLS
      END IF
      IF (TOTWELL.LT.TOT(NWLLS)) TOTWELL=TOT(NWLLS)
      DO 60 IADD=1,5
        READ (5,40,END=100) LINE
        ADDRESS(NWLLS,IADD)=LINE
   60 CONTINUE
      INDEX=0
      K=0
   70 K=K+1
      READ (5,40,END=100) LINE
      DO 80 L=5,61,14
        INDEX=INDEX+1
        INPT=LINE(L:L+9)
        CALL GETNO (INPT,DBDATA(NWLLS,INDEX),DBSFG(NWLLS,INDEX),ERR)
        IF (ERR.GT.0) GO TO (120,140,160),ERR
   80 CONTINUE
      IF (K.LT.9) GO TO 70
      READ (5,40) LINE
      FORMATION(NWLLS)=LINE(1:17)
      GO TO 10
   90 CLOSE (5)
      RETURN
  100 WRITE (*,110)
  110 FORMAT (/' **** Error ****'/' End of file before all data in last
     *well was read.'/' ***************'/)
      STOP
  120 WRITE (*,130) NWLLS,INDEX,WLLNMS(NWLLS)(1:79)
  130 FORMAT (/' **** Error ****'/' Reading error at well#',I3,' index#'
```

```
     *,I3/1X,A79/'***************'/)
      STOP
  140 WRITE (*,150) NWLLS,INDEX,WLLNMS(NWLLS)(1:79)
  150 FORMAT (/' **** Error ****'/' No decimal point in number.'/' well#
     *',I3,' index#',I3/1X,A79/'***************'/)
      STOP
  160 WRITE (*,170) NWLLS,INDEX,WLLNMS(NWLLS)(1:79)
  170 FORMAT (/' **** Error ****'/' Unknown error at well#',I3,' index#'
     *,I3/1X,A79/'***************'/)
      STOP
  180 WRITE (*,190)
  190 FORMAT (/' **** Error ****'/' Arrays not big enough to hold all of
     * data base.'/' ***************'/)
      STOP
      END
      SUBROUTINE SAVE
      CHARACTER ANS
   10 WRITE (*,20)
      WRITE (*,30)
   20 FORMAT (' (1) Save raw data, (2) create .PATH file, or (3) both?')
   30 FORMAT (' (<Enter> for both, 0 to abort)')
   40 FORMAT (A)
   50 FORMAT (I1)
      READ (*,40,ERR=10,END=10) ANS
      READ (ANS,50,ERR=10) I
      IF (ANS.EQ.' ') I=3
      IF (I.EQ.1.OR.I.EQ.3) CALL SAVELONG
      IF (I.EQ.2.OR.I.EQ.3) CALL SAVEPATH
      WRITE (*,40) ' Hit <Enter> to continue'
      READ (*,40) ANS
      RETURN
      END
      SUBROUTINE SAVELONG
      CHARACTER*80 WLLNMS(50),LINE
      CHARACTER*40 ADDRESS(50,5),LAT(50),DFILE
      CHARACTER*17 FORMATION(50)
      CHARACTER*7 FRMT
      CHARACTER*10 PIECE
      DIMENSION DBDATA(50,45), PRNT(5)
      INTEGER COUNT,DBSFG(50,45),NWLLS,STYLE,TOTWELL,TOT(50)
      COMMON /DB/ DBDATA,DBSFG,NWLLS,WLLNMS,ADDRESS,LAT,FORMATION,IU(50,
     *4),IDEFAULT(5),TOTWELL,TOT
      COMMON /FILE/ DFILE,ISAVE
      WRITE (*,*) 'Saving .LONG...'
      OPEN (6,FILE=DFILE//'.LON')
   10 FORMAT (A)
      DO 110 N=1,NWLLS
        WRITE (6,20) IU(N,1),IU(N,2),IU(N,3),IU(N,4),WLLNMS(N)(5:80)
   20   FORMAT (4(I1),A76)
        WRITE (6,30) LAT(N),TOT(N),TOTWELL
   30   FORMAT (A40,24X,'#',I5,' of ',I5)
        DO 40 I=1,5
   40     WRITE (6,10) ADDRESS(N,I)
        JCOUNTER=41
        DO 90 I=1,JCOUNTER,5
          LINE=' '
          DO 80 K=0,4
            J=I+K
            IF (DBSFG(N,J).LT.0) GO TO 50
            FRMT='(F10.'//CHAR(ICHAR('0')+DBSFG(N,J))//')'
            WRITE (PIECE,FRMT) DBDATA(N,J)
```

```
            GO TO 80
   50       IF (DBSFG(N,J).EQ.-2) GO TO 60
            PIECE='**********'
            GO TO 80
   60       WRITE (PIECE,70) DBDATA(N,J)
   70       FORMAT ('<',F9.3)
   80       WRITE (LINE,10) LINE(1:K*14+4)//PIECE//LINE(K*14+15:)
   90     WRITE (6,10) LINE
  100   FORMAT (A17)
        WRITE (6,100) FORMATION(N)
  110 CONTINUE
      CLOSE (6)
      RETURN
      END
      SUBROUTINE SAVEPATH
      CHARACTER*80 WLLNMS(50),LINE,OUTFORM
      CHARACTER*40 ADDRESS(50,5),LAT(50),DFILE
      CHARACTER*17 FORMATION(50)
      CHARACTER*10 FMT
      CHARACTER SFG
      DIMENSION DBDATA(50,45), PRNT(5), DAT(45)
      INTEGER COUNT,DBSFG(50,45),NWLLS,STYLE,TOTWELL,TOT(50),PS(8)
      COMMON /DB/ DBDATA,DBSFG,NWLLS,WLLNMS,ADDRESS,LAT,FORMATION,IU(50,
     *4),IDEFAULT(5),TOTWELL,TOT
      COMMON /FILE/ DFILE,ISAVE
C THESE DATA WILL BE SENT DIRECTLY TO PATH:
      DATA PS/29,30,36,33,34,5,31,32/
      ISAVE=1
      WRITE (*,10)
   10 FORMAT (/,' Do you want to adjust the .PATH file for possible',' c
     *harge imbalance?',/,' <Enter> = no')
      READ (*,20) LINE
   20 FORMAT (A1)
      ICORT=0
      IF (LINE.EQ.'Y'.OR.LINE.EQ.'y') ICORT=1
      WRITE (*,*) 'Saving WATEQF input file...'
      OPEN (6,FILE=DFILE//'.IN')
      DO 140 I=1,NWLLS
        DO 30 JJ=1,45
          DAT(JJ)=DBDATA(I,JJ)
   30     IF (DBSFG(I,JJ).LT.0) DAT(JJ)=0
C HANDLE DEFAULT DENSITY (1, NOT 0)
        IF (DAT(28).EQ.0.) DAT(28)=1.
        WRITE (6,40) WLLNMS(I)(5:36)
   40   FORMAT ('''',A,'''')
C FIND VALUE FOR EH
        EHM=DAT(8)
        IPE=IU(I,2)
        IF (IPE.NE.1) EHM=9.9
        IF (IPE.EQ.4.AND.DAT(6)*DAT(13).LE.0) IPE=0
        IF (IPE.EQ.0.AND.(DAT(21).GT.0.OR.DAT(22).GT.0)) WRITE (*,50)
     *   WLLNMS(I)(5:36)
   50   FORMAT (' WARNING - Eh set to 0 for ',A)
        IF (IPE.EQ.0) THEN
          IPE=1
          EHM=0.0
        END IF
        IF (DAT(5).LT.0.AND.DBSFG(I,5).GE.0) THEN
          DAT(5)=0
          WRITE (*,60) WLLNMS(I)(5:36)
        END IF
```

```
60    FORMAT (' Tritium negative: saved as 0 in .PAT file: ',A)         AP 480
      DOX=DAT(3)                                                        AP 490
      WRITE (6,70) DAT(1),DAT(2),EHM,DAT(28),DOX,IU(I,1),IU(I,3),IPE,   AP 500
     * IU(I,4)                                                          AP 510
70    FORMAT (F6.3,1X,F6.3,1X,F4.2,' 9.9 9.9 ',F7.5,F6.2,3(I2),5(' 0') AP 520
     * ,I2,' 0 0')                                                      AP 530
      OUTFORM='(F10.3,1X,F10.3,1X,F10.3,1X,F10.3,1X,F10.3,1X,F10.3)'    AP 540
      IF (DBSFG(I,7).GE.0) WRITE (OUTFORM,80) OUTFORM(1:5),DBSFG(I,7),  AP 550
     * OUTFORM(7:)                                                      AP 560
      IF (DBSFG(I,9).GE.0) WRITE (OUTFORM,80) OUTFORM(1:14),DBSFG(I,9) AP 570
     * ,OUTFORM(16:)                                                    AP 580
      IF (DBSFG(I,10).GE.0) WRITE (OUTFORM,80) OUTFORM(1:23),DBSFG(I,  AP 590
     * 10),OUTFORM(25:)                                                 AP 600
      IF (DBSFG(I,11).GE.0) WRITE (OUTFORM,80) OUTFORM(1:32),DBSFG(I,  AP 610
     * 11),OUTFORM(34:)                                                 AP 620
      IF (DBSFG(I,12).GE.0) WRITE (OUTFORM,80) OUTFORM(1:41),DBSFG(I,  AP 630
     * 12),OUTFORM(43:)                                                 AP 640
      IF (DBSFG(I,13).GE.0) WRITE (OUTFORM,80) OUTFORM(1:50),DBSFG(I,  AP 650
     * 13),OUTFORM(52:)                                                 AP 660
      WRITE (6,OUTFORM) DAT(7),DAT(9),DAT(10),DAT(11),DAT(12),DAT(13)  AP 670
80    FORMAT (A,I1,A)                                                   AP 680
      OUTFORM='(F10.3,1X,F10.3,1X,F10.3,1X,F10.3,1X,F10.3,1X,F10.3)'    AP 690
      IF (DBSFG(I,4).GE.0) WRITE (OUTFORM,80) OUTFORM(1:5),DBSFG(I,4),  AP 700
     * OUTFORM(7:)                                                      AP 710
      IF (DBSFG(I,15).GE.0) WRITE (OUTFORM,80) OUTFORM(1:14),DBSFG(I,  AP 720
     * 15),OUTFORM(16:)                                                 AP 730
      IF (DBSFG(I,21).GE.0) WRITE (OUTFORM,80) OUTFORM(1:23),DBSFG(I,  AP 740
     * 21),OUTFORM(25:)                                                 AP 750
      IF (DBSFG(I,25).GE.0) WRITE (OUTFORM,80) OUTFORM(1:32),DBSFG(I,  AP 760
     * 25),OUTFORM(34:)                                                 AP 770
      IF (DBSFG(I,20).GE.0) WRITE (OUTFORM,80) OUTFORM(1:41),DBSFG(I,  AP 780
     * 20),OUTFORM(43:)                                                 AP 790
      IF (DBSFG(I,14).GE.0) WRITE (OUTFORM,80) OUTFORM(1:50),DBSFG(I,  AP 800
     * 14),OUTFORM(52:)                                                 AP 810
      WRITE (6,OUTFORM) DAT(4),DAT(15),DAT(21),DAT(25),DAT(20),DAT(14) AP 820
      OUTFORM='(''''''CONC'''''','' 17 '',F10.3,'' 98 '',F10.3,'' 87 ' AP 830
     *',F10.3,'' 90 '',F10.3,'' 81 '',F10.3)'                          AP 840
      IF (DBSFG(I,6).GE.0) WRITE (OUTFORM,80) OUTFORM(1:23),DBSFG(I,6) AP 850
     * ,OUTFORM(25:)                                                    AP 860
      IF (DBSFG(I,16).GE.0) WRITE (OUTFORM,80) OUTFORM(1:36),DBSFG(I,  AP 870
     * 16),OUTFORM(38:)                                                 AP 880
      IF (DBSFG(I,17).GE.0) WRITE (OUTFORM,80) OUTFORM(1:49),DBSFG(I,  AP 890
     * 17),OUTFORM(51:)                                                 AP 900
      IF (DBSFG(I,18).GE.0) WRITE (OUTFORM,80) OUTFORM(1:62),DBSFG(I,  AP 910
     * 18),OUTFORM(64:)                                                 AP 920
      IF (DBSFG(I,19).GE.0) WRITE (OUTFORM,80) OUTFORM(1:75),DBSFG(I,  AP 930
     * 19),OUTFORM(77:)                                                 AP 940
      WRITE (6,OUTFORM) DAT(6),DAT(16),DAT(17),DAT(18),DAT(19)         AP 950
      OUTFORM='(''''''CONC'''''','' 101 '',F10.3,'' 85 '',F10.3,'' 39  AP 960
     *'',F10.3,'' 51 '',F10.3,'' 116 '',F10.3)'                        AP 970
      IF (DBSFG(I,22).GE.0) WRITE (OUTFORM,80) OUTFORM(1:24),DBSFG(I,  AP 980
     * 22),OUTFORM(26:)                                                 AP 990
      IF (DBSFG(I,23).GE.0) WRITE (OUTFORM,80) OUTFORM(1:37),DBSFG(I,  AP1000
     * 23),OUTFORM(39:)                                                 AP1010
      IF (DBSFG(I,24).GE.0) WRITE (OUTFORM,80) OUTFORM(1:50),DBSFG(I,  AP1020
     * 24),OUTFORM(52:)                                                 AP1030
      IF (DBSFG(I,37).GE.0) WRITE (OUTFORM,80) OUTFORM(1:63),DBSFG(I,  AP1040
     * 37),OUTFORM(65:)                                                 AP1050
      IF (DBSFG(I,35).GE.0) WRITE (OUTFORM,80) OUTFORM(1:77),DBSFG(I,  AP1060
     * 35),OUTFORM(79:)                                                 AP1070
      WRITE (6,OUTFORM) DAT(22),DAT(23),DAT(24),DAT(37),DAT(35)        AP1080
```

```
        OUTFORM='(''''''CONC'''''','' 117 '',F10.3,'' 118 '',F10.3      AP1090
     *  ,'' 0 0 0 0 0 0  '')'                                           AP1100
        IF (DBSFG(I,26).GE.0) WRITE (OUTFORM,80) OUTFORM(1:24),DBSFG(I, AP1110
     *   26),OUTFORM(26:)                                               AP1120
        IF (DBSFG(I,45).GE.0) WRITE (OUTFORM,80) OUTFORM(1:38),DBSFG(I, AP1130
     *   45),OUTFORM(40:)                                               AP1140
        WRITE (6,OUTFORM) DAT(26),DAT(45)                               AP1150
        WRITE (6,90)                                                    AP1160
   90   FORMAT (''' '' 0 0 0 0 0 0 0 0 0 0')                            AP1170
        LINE=' '                                                        AP1180
        II=0                                                            AP1190
  100   II=II+1                                                         AP1200
        FMT='(F8.1,'' '')'                                              AP1210
        IF (DBSFG(I,PS(II)).GE.0) THEN                                  AP1220
          WRITE (FMT,'(A4,I1,A5)') FMT(1:4),DBSFG(I,PS(II)),FMT(6:10)   AP1230
        ELSE                                                            AP1240
          IF (DBSFG(I,PS(II)).EQ.-1) WRITE (FMT,'(A7,A1,A2)') FMT(1:7),' AP1250
     **',FMT(9:10)                                                      AP1260
        END IF                                                          AP1270
        WRITE (LINE(9*II-8:9*II),FMT) DAT(PS(II))                       AP1280
        IF (II.EQ.8) GO TO 110                                          AP1290
        GO TO 100                                                       AP1300
  110   WRITE (LINE,120) LINE(1:74),TOT(I),ICORT                        AP1310
  120   FORMAT (A74,I5,I1)                                              AP1320
        WRITE (6,'(A80)') LINE                                          AP1330
        LINE=' '                                                        AP1340
        DO 130 KK=1,45                                                  AP1350
          SFG=' '                                                       AP1360
          IF (DBSFG(I,KK).EQ.-1) SFG='*'                                AP1370
          WRITE (LINE,'(A,A1,A)') LINE(:KK-1),SFG,LINE(KK+1:)           AP1380
  130   CONTINUE                                                        AP1390
        WRITE (6,'(A80)') LINE                                          AP1400
  140 CONTINUE                                                          AP1410
      CLOSE (6)                                                         AP1420
      RETURN                                                            AP1430
      END                                                               AP1440
      SUBROUTINE SWITCH (S,D,*)                                         AQ  10
      CHARACTER*80 LINE,WLLNMS(50)                                      AQ  20
      CHARACTER*17 FORMATION(50)                                        AQ  30
      CHARACTER*40 ADDRESS(50,5),LAT(50)                                AQ  40
      DIMENSION DBDATA(50,45)                                           AQ  50
      INTEGER D,DBSFG(50,45),NWLLS,S,TOTWELL,TOT(50)                    AQ  60
      COMMON /DB/ DBDATA,DBSFG,NWLLS,WLLNMS,ADDRESS,LAT,FORMATION,IU(50,AQ  70
     *4),IDEFAULT(5),TOTWELL,TOT                                        AQ  80
      CALL MOVE (S,NWLLS+1)                                             AQ  90
      IF (D-S) 10,30,40                                                 AQ 100
   10 DO 20 I=S-1,D,-1                                                  AQ 110
   20   CALL MOVE (I,I+1)                                               AQ 120
      CALL MOVE (NWLLS+1,D)                                             AQ 130
      RETURN                                                            AQ 140
   30 RETURN 1                                                          AQ 150
   40 DO 50 I=S+1,D                                                     AQ 160
   50   CALL MOVE (I,I-1)                                               AQ 170
      CALL MOVE (NWLLS+1,D)                                             AQ 180
      RETURN                                                            AQ 190
      END                                                               AQ 200
      CHARACTER*40 FUNCTION UPCS(LINE)                                  AR  10
C                                                                       AR  20
C        This subroutine changes all the letters in a                   AR  30
C        line to upper case.                                            AR  40
C                                                                       AR  50
```

```
      CHARACTER*(*) LINE
      UPCS=LINE
      DO 10 I=1,LEN(LINE)
        ICH=ICHAR(LINE(I:I))
   10   IF (ICH.GE.ICHAR('a').AND.ICH.LE.ICHAR('z')) UPCS(I:I)=CHAR(ICH-
     *   ICHAR('a')+ICHAR('A'))
      RETURN
      END
      SUBROUTINE WELLFILE
C
C The well data file to be used is selected here.
C
      CHARACTER*80 FILES(100),OFILES(100),LINE
      CHARACTER*40 DFILE,UPCS
      COMMON /FILE/ DFILE,ISAVE
      ISAVE=0
      NFILES=0
      OPEN (UNIT=7,FILE='DB.FIL',STATUS='OLD',ERR=40)
      REWIND (7)
   10 READ (7,130,ERR=10,END=40) LINE
      DO 20 I=34,1,-1
        IF (LINE(I:I).NE.' ') GO TO 30
   20 CONTINUE
      GO TO 10
   30 LINE(I+1:)='.LON'
      OPEN (8,FILE=LINE,STATUS='OLD',ERR=10)
      CLOSE (8)
      NFILES=NFILES+1
      FILES(NFILES)=UPCS(LINE(1:I))
      GO TO 10
   40 IF (NFILES.EQ.0) GO TO 140
   50 CLOSE (7)
      CLOSE (8)
      IJ=1
   60 ICOUNT=(IJ-1)*15
      CALL CLS
      WRITE (*,70)
   70 FORMAT (' ----------------',/,' CHOOSE DATA FILE',/,' ------------
     *----',/)
   80 ICOUNT=ICOUNT+1
   90 FORMAT (I4,':  ',A40)
      WRITE (*,90) ICOUNT,FILES(ICOUNT)
      IF (ICOUNT.LT.NFILES.AND.ICOUNT.LT.IJ*15) GO TO 80
      DO 100 I=ICOUNT,(IJ*15-1)
        WRITE (*,*)
  100 CONTINUE
      IF (IJ*15.LT.NFILES) THEN
        IJ=IJ+1
      ELSE
        IJ=1
      END IF
      IF (NFILES.LE.15) WRITE (*,110)
      IF (NFILES.GT.15) WRITE (*,120)
  110 FORMAT (/,' Enter number of file to use, or <Enter> to continue:')
  120 FORMAT (/,' Enter number of file to use, ''M'' to see more',' choi
     *ces, or <ENTER> to continue')
```

```
      READ (*,130) LINE                                                 AS 490
  130 FORMAT (A80)                                                      AS 500
      IF (LINE.EQ.' ') GO TO 140                                        AS 510
      IF (UPCS(LINE).EQ.'M') GO TO 60                                   AS 520
      READ (LINE,'(I3)',ERR=60) I                                       AS 530
      IF (I.LT.1.OR.I.GT.NFILES) GO TO 50                               AS 540
      GO TO 160                                                         AS 550
  140 CALL CLS                                                          AS 560
      WRITE (*,150)                                                     AS 570
      READ (*,130) LINE                                                 AS 580
      IF (LINE.EQ.' ') STOP                                             AS 590
      DFILE=LINE                                                        AS 600
      LINE(LENS(LINE)+1:)='.LON'                                        AS 610
      OPEN (UNIT=9,FILE=LINE,ERR=140)                                   AS 620
      CLOSE (9)                                                         AS 630
      NFILES=NFILES+1                                                   AS 640
      FILES(NFILES)=DFILE                                               AS 650
      GO TO 170                                                         AS 660
  150 FORMAT (//,' Enter file prefix to use, or <Enter> to quit:')      AS 670
  160 DFILE=FILES(I)                                                    AS 680
  170 OPEN (UNIT=7,FILE='DB.FIL')                                       AS 690
      CLOSE (7,STATUS='DELETE')                                         AS 700
      OPEN (UNIT=7,FILE='DB.FIL',STATUS='NEW')                          AS 710
      DO 180 I=1,NFILES                                                 AS 720
        WRITE (7,'(A)') FILES(I)(1:LENS(FILES(I)))                      AS 730
  180 CONTINUE                                                          AS 740
      CLOSE (7)                                                         AS 750
      RETURN                                                            AS 760
      END                                                               AS 770
C     WATEQFP.FOR                                                       AT  10
C     **** PROGRAM WATEQF.PATH **** A FORTRAN IV VERSION OF WATEQ       AT  20
C     FOR USE IN CREATING INPUT FILES FOR NETPATH                       AT  30
C                                                                       AT  40
C     REVISED FROM PL1 VERSION OF TRUESDELL AND JONES.                  AT  50
C     NIEL PLUMMER, SUMMER 1972.                                        AT  60
C     LATEST REVISION: ERIC PRESTEMON: MARCH, 1991                      AT  70
C                                                                       AT  80
C                                                                       AT  90
C     **** DESCRIPTION OF INPUT - 5 CARDS ARE REQUIRED ****             AT 100
C     CARD 1   TITLE, JOB DESCRIPTION.  (A80)                           AT 110
C     CARD 2    TEMP,PH,EHM,EHMC,EHMZ,DENS,DOX,FLAG,CORALK,PECALC,IGO,  AT 120
C               (PRT(I),I=1,4),IDAVES,ISPEC,IMIN                        AT 130
C               (5(F6.0,1X),2F5.0,1X,9I1,2I3)                           AT 140
C       TEMP....TEMPERATURE IN DEGREES C                                AT 150
C       PH......NEGATIVE LOG ACTIVITY H+                                AT 160
C       EHM.....PREFERRED EH ...SEE OPTIONS                             AT 170
C       EHMC....MEASURED EH ... SEE OPTIONS                             AT 180
C       EMFZ....MEASURED EH OF ZOBELL SOLUTION                          AT 190
C       DENS....DENSITY OF SOLUTION   (G/CC)                            AT 200
C       DOX.....DISSOLVED OXYGEN (MG/L)                                 AT 210
C       FLAG....SIGNAL FOR UNITS OF INPUT CONCENTRATION.                AT 220
C         0 (OR BLANK) = MMOLE/L, 1=MEQ/L, 2=MG/L, 3=PPM, 4=MOLALITY.   AT 230
C       CORALK..=0 IF ALKALINITY HAS NOT BEEN CORRECTED FOR BORON ETC.  AT 240
C         AND THE ORIGINAL EXPRESSION OF WATEQ IS TO BE USED.  =1 IF    AT 250
C         CARBONATE ALKALINITY (CORRECTED FOR NON-CARBONATE ALKALINITY  AT 260
C         SPECIES) HAS BEEN INPUT. =2 IF TOTAL INORGANIC CARBON IS      AT 270
C         INPUT RATHER THAN ALKALINITY.  =3 IF ALKALINITY HAS NOT BEEN  AT 280
C         CORRECTED FRO BORON ETC. - SIMILAR TO CORALK=0, EXCEPT THAT   AT 290
C         ALL POSSIBLE NON-CARBONATE ALKALINITY SPECIES ARE CONSIDERED. AT 300
C       PECALC..=0 WILL SET PE TO 100,  =1 COMPUTES PE FROM EH,         AT 310
C         =2 COMPUTES PE FROM DOX(THEORETICAL).  = 3 COMPUTES PE FROM   AT 320
```

```
C         THE SATO RELATION,  = 4 COMPUTES PE FROM S-- - SO4--.          AT 330
C       IGO..=0,OR BLANK, IF DESIRED TO HAVE DATA CHECKED FOR INPUT      AT 340
C         ERROR. PH MUST BE GREATER THAN 3 AND LESS THAN 11, AND THE     AT 350
C         ANALYSIS MUST HAVE LESS THAN 30% ERROR IN CHARGE BALANCE.  =1  AT 360
C         IF THIS CHECK IS NOT TO BE MADE.                               AT 370
C       (PRT(I),I=1,4), CAN BE SET TO 1 TO DELETE PRINT OF               AT 380
C          THERMOCHEMICAL DATA,MASS BALANCE CONVERGENCE ITERATIONS,      AT 390
C         RATIOS OF IONS, AND MINERAL SATURATION, RESPECTIVELY.  PRT(I)  AT 400
C         SHOULD BE SET TO ZERO OR BLANK TO OBTAIN THE RESPECTIVE PRINT. AT 410
C       IDAVES..=1, ACTIVITY COEFFICIENTS OF CHARGED ION PAIRS ARE       AT 420
C         CALCULATED FROM THE DAVIES EQUATION.  =0 (OR BLANK), ACTIVITY  AT 430
C         COEFFICIENTS OF CHARGED ION PAIRS ARE CALCULATED FROM THE      AT 440
C         DEBYE-HUCKEL EQUATION.  IDAVES HAS NO EFFECT ON GAMMA(1)-      AT 450
C         GAMMA(7), AND GAMMA(18).                                       AT 460
C       ISPEC.. = NUMBER OF SPECIES DESIRED IN OUTPUT(IF LESS THAN TOTAL AT 470
C         POSSIBLE). TO OBTAIN OUTPUT OF MOLALITY, ACTIVITY, ETC. OF     AT 480
C         ALL POSSIBLE SPECIES FOR THE DEFINED SYSTEM, LEAVE ISPEC       AT 490
C         BLANK (OR ZERO).  IF ISPEC GT. ZERO, ISPEC VALUES OF KSPEC     AT 500
C         (SPECIES INDEX NUMBER) MUST BE READ (SEE BELOW).  IF ISPEC =   AT 510
C         BLANK (ZERO), OMIT KSPEC CARD(S).                              AT 520
C       IMIN.. = NUMBER OF MINERALS FOR WHICH SATURATION OUTPUT IS       AT 530
C         DESIRED (IF LESS THAN TOTAL POSSIBLE).  TO OBTAIN SATURATION   AT 540
C         DATA ON ALL POSSIBLE MINERALS FOR THE DEFINED SYSTEM, LEAVE    AT 550
C         IMIN BLANK (OR ZERO).  IF IMIN GT. ZERO, IMIN VALUES OF KMIN   AT 560
C         (MINERAL INDEX NUMBER) MUST BE READ (SEE BELOW).  IF IMIN =    AT 570
C         BLANK (OR ZERO), OMIT KMIN CARDS(S).                           AT 580
C     CARD 3    CA MG NA K CL SO4    (6(E12.5),8X)                       AT 590
C     CARD 4    HCO3 SIO2 FE PO4 (AS P) SR F      (6(E12.5),8X)          AT 600
C             ... OPTIONAL CARDS OF TYPE 1 APPEAR HERE ...               AT 610
C             ... OPTIONAL CARDS OF TYPE 2 APPEAR HERE ..                AT 620
C     CARD 5    BLANK CARD     (DENOTES END OF DATA FOR A PARTICULAR     AT 630
C         WATER ANALYSIS.)                                               AT 640
C                                                                        AT 650
C     ....DESCRIPTION OF OPTIONAL INPUT....                              AT 660
C         ALL OPTIONAL INPUT MUST APPEAR BETWEEN CARDS 4 AND 5.          AT 670
C         TYPE 1 CARDS MUST PRECEED TYPE 2 CARDS.                        AT 680
C                                                                        AT 690
C     ***************************                                        AT 700
C     TYPE 1 OPTIONAL INPUT CARDS                                        AT 710
C     ***************************                                        AT 720
C     (KSPEC(I),I=1,ISPEC)  (16I5)  KSPEC(I) IS THE INDEX NUMBER OF THE  AT 730
C         ITH SELECTED SPECIES FOR WHICH OUTPUT IS DESIRED.  OMIT CARD   AT 740
C         IF ISPEC = BLANK (OR ZERO).                                    AT 750
C     (KMIN(I),I=1,IMIN)  (16I5)  KMIN(I) IS THE INDEX NUMBER OF THE     AT 760
C         ITH SELECTED MINERAL FOR WHICH SATURATION OUTPUT IS DESIRED.   AT 770
C         OMIT CARD IF IMIN = BLANK (OR ZERO).                           AT 780
C     NOTE THAT IF BOTH KSPEC AND KMIN ARE READ, KSPEC(I) MUST BE READ   AT 790
C     BEFORE KMIN(I).                                                    AT 800
C                                                                        AT 810
C     ***************************                                        AT 820
C     TYPE 2 OPTIONAL INPUT CARDS                                        AT 830
C     ***************************                                        AT 840
C     WORD,(INT(I),VAL(I),I=1,5)  (A4,1X,5(I3,E12.5))                    AT 850
C     WORD = 'CONC', 'EROR', 'DELH', 'TABL', OR'LOGK'.                   AT 860
C                                                                        AT 870
C       'CONC'..ENTERS CONCENTRATION (UNITS OF FLAG) OF CONSTITUENTS     AT 880
C       NOT ON CARDS 3 AND 4.  INT(I) = 17(H2S),18(CO3),39(NH4 as N),    AT 890
C       51(AL),81(LI),85(NO3 as N),86(H2CO3),87(B),90(BA),98(BR),        AT 900
C       AND 101(MN).                                                     AT 910
C ***NOTE: (as N) for 39 and 85 is a change!                             AT 920
C       VAL(I) IS THE CONCENTRATION OF THE INT(I) CONSTITUENT.           AT 930
```

```
C                                                                       AT 940
C       'EROR'..OVERRIDES PRE-SET MASS BALANCE CONVERGENCE CONSTRAINTS  AT 950
C       ON ANIONS.  PER-SET VALUES OF EROR1-EROR5 ARE 0.001(0.1% ERROR  AT 960
C       IN MASS BALANCE).  EROR1-EROR5 ARE ENTERED ON THE 'EROR' CARD   AT 970
C       AS VAL(1)-VAL(5), IN THE ORDER 1=CARBON, 2=SULFATE, 3=FLUORIDE, AT 980
C       4=PHOSPHATE, 5=CHLORIDE.  VALUES OF INT(I) ARE NOT USED.        AT 990
C                                                                       AT1000
C       'DELH'..OVERRIDES VALUES OF THE STANDARD DELTA ENTHALPY OF      AT1010
C       REACTION (25 DEG. C) USED IN COMPUTING THE TEMPERATURE          AT1020
C       DEPENDENCE OF EQUILIBRIUM CONSTANTS FROM THE VANT HOFF EQUATION. AT1030
C       INT(I) IS THE INDEX NUMBER OF THE ITH REACTION FOR WHICH DH(I)  AT1040
C       IS TO BE CHANGED AND VAL(I) IS THE APPPRIATE NEW VALUE OF       AT1050
C       DH(INT(I)).                                                     AT1060
C                                                                       AT1070
C       'TABL'..OVERRIDES VALUES OF LOGKTO(INT(I)) (LOG K OF REACTION AT AT1080
C       25 DEG. C USED IN COMPUTING THE TEMPERATURE DEPENDENCE OF       AT1090
C       EQUILIBRIUM CONSTANTS FROM THE VANT HOFF EQUATION).  INT(I) IS  AT1100
C       THE INDEX NUMBER OF THE ITH REACTION FOR WHICH LOGKTO IS TO BE  AT1110
C       CHANGED AND VAL(I) IS THE APPROPRIATE NEW VALUE OF LOGKTO(I).   AT1120
C                                                                       AT1130
C       'LOGK'..OVERRIDES EXISTING ANALYTICAL EXPRESSIONS FOR LOG K AS A AT1140
C       FUNCTION OF T(DEG.K), OR ENTERS NEW, PREVIOUSLY UNDEFINED       AT1150
C       ANALYTICAL EXPRESSIONS FOR LOG K(T DEG.K).  SIX VALUES ARE      AT1160
C       READ, INSTEAD OF THE NORMAL FIVE.  THE FORM OF THE              AT1170
C       ANALYTICAL EXPRESSION MUST BE                                   AT1180
C             LOG KT(INT(I))=A+B*T+C/T+D*T**2+E/T**2+F*LOG(T)           AT1190
C       WHERE T IS TEMPERATURE IN DEG. K, AND A,B,C,D,E AND F ARE FIT   AT1200
C       PARAMETERS (MAY BE ZERO OR BLANK).  INT(1) IS THE INDEX NUMBER  AT1210
C       OF REACTION AND INT(2)-INT(5) ARE IGNORED.  VAL(1)=A,VAL(2)=B,  AT1220
C       VAL(3)=C,VAL(4)=D,VAL(5)=E,VAL(6)=F.                            AT1230
C                                                                       AT1240
C                                                                       AT1250
C     IF ANY OF THE CARDS, 'EROR','DELH','TABL','LOGK', ARE USED IN A   AT1260
C     PARTICULAR WATER DATA SET, CALCULATIONS FOR THAT DATA SET AND ALL AT1270
C     SUBSEQUENT DATA SETS WILL USE THE NEW INPUT VALUES.  DELH AND TABL AT1280
C     CARDS CAN BE USED TO OVERRIDE PRE-EXISTING ANALYTICAL EXPRESSIONS. AT1290
C     THE ORDER OF TYPE 2 OPTIONAL INPUT CARDS IS 'CONC','EROR','DELH', AT1300
C     'TABL',AND 'LOGK', IF ALL 5 ARE USED.  THE LAST CARD IN EACH WATER AT1310
C     ANALYSIS DATA SET MUST BE BLANK.                                  AT1320
C                                                                       AT1330
      SUBROUTINE WATEQFP                                                AT1340
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               AT1350
      INTEGER D,E,DD,RBIT,CORALK,Z(120),PRT(4)                          AT1360
      INTEGER PECALC,PECK,ICK(2)                                        AT1370
      DOUBLE PRECISION MI(120),KT(200),LOGKT(200),LOGKTO(200),MNTOT,    AT1380
     *LH2O,MU,NATOT,KTOT,MGTOT,LITOT,NH4TOT,KW                          AT1390
      CHARACTER*8 NSPEC(120),NREACT(200)                                AT1400
      CHARACTER*80 TITL,PATHLINE,PL2                                    AT1410
      CHARACTER*40 DFILE                                                AT1420
      CHARACTER ANSWER                                                  AT1430
      COMMON MI,KT,LOGKT,LOGKTO,KW,D,E,DD,C,R,T,F,TEMP,A,B,PE,PES,PEDO, AT1440
     *PESATO,PECK,PECALC,PH,TENMPE,TENPH,ALFA(120),GAMMA(120),AP(200),  AT1450
     *XLALFA(120),Z,CUNITS(120),ANALMI(120),GFW(120),DHA(120),NSPEC,    AT1460
     *NREACT,DH(200),AH2O,LH2O,EROR1,EROR2,EROR3,EROR4,EROR5,EHM,DENS,  AT1470
     *DOX,XLMI(120),ITER,RBIT,C1SAVE,CORALK,MU,LCHEK(200),CO2TIT,ANALCO, AT1480
     *SITOT,CATOT,MGTOT,KTOT,NATOT,SO4TOT,FETOT,PTOT,ALTOT,FTOT,BTOT,   AT1490
     *LITOT,NH4TOT,SRTOT,BATOT,CLTOT,MNTOT,ICK,PRT,TITL,EPMCAT,EPMAN,   AT1500
     *NEQU,ISPEC,KSPEC(120),IMIN,KMIN(200),TDS,IDAVES,IPRT,JJ,JK,       AT1510
     *PATHLINE,PL2,IMBAL                                                AT1520
      COMMON /FILE/ DFILE,ISAVE                                         AT1530
      IF (ISAVE.EQ.0) RETURN                                            AT1540
```

```
   WRITE (*,*) ' Running WATEQF - Please wait.'                       AT1550
   JJ=0                                                               AT1560
   JK=0                                                               AT1570
   D=115                                                              AT1580
   E=193                                                              AT1590
   IPRT=0                                                             AT1600
   NEQU=22                                                            AT1610
   OPEN (UNIT=5,FILE=DFILE//'.IN')                                    AT1620
   OPEN (UNIT=6,FILE=DFILE//'.OUT')                                   AT1630
   OPEN (UNIT=7,FILE=DFILE//'.PAT',STATUS='UNKNOWN')                  AT1640
   OPEN (UNIT=9,FILE='WATEQF.DAT')                                    AT1650
   OPEN (UNIT=11,FILE='TEMP',STATUS='UNKNOWN')                        AT1660
   CLOSE (UNIT=11,STATUS='DELETE')                                    AT1670
   IMBAL=1                                                            AT1680
   IF (I.GE.1.AND.I.LE.3) IMBAL=I                                     AT1690
   REWIND 7                                                           AT1700
   WRITE (7,10)                                                       AT1710
10 FORMAT ('ORDER IS:  WELL,',/,'C, S, CA, AL, MG, NA, K, CL,',/,'F,  AT1720
  *SI, BR, B, BA, LI, SR, FE,',/,'MN, N, P, TEMP, H2S, SO4, NO3, NH4, AT1730
  *',/,'DOX, HCO3, PH, H2CO3*, CO3, CARBONATES, FE2+, FE3+',/,' MN2+, AT1740
  * MN3+, MN6+, MN7+, CH4, DOC, RS OF DOC, BLANK'/,'C13, C14, SR87, D AT1750
  *, O-18, TRITIUM, 34SSO4, S4SH2S',/)                                AT1760
   DO 20 I=1,D                                                        AT1770
20   READ (9,*) NSPEC(I),Z(I),GFW(I),DHA(I)                           AT1780
   DO 30 I=1,E                                                        AT1790
30   READ (9,*) NREACT(I),DH(I),LOGKTO(I)                             AT1800
   ICK(2)=0                                                           AT1810
40 READ (5,*,END=100) TITL                                            AT1820
   CALL PREP                                                          AT1830
   IF (ICK(1).EQ.1) WRITE (*,70) TITL                                 AT1840
   IF (ICK(1).EQ.1.AND.ICK(2).EQ.0) THEN                              AT1850
     WRITE (*,50)                                                     AT1860
     READ (*,60) ANSWER                                               AT1870
     ICK(2)=1                                                         AT1880
     IF (ANSWER.EQ.'N'.OR.ANSWER.EQ.'n') ICK(2)=2                     AT1890
   END IF                                                             AT1900
50 FORMAT (' Do you want to make .PATH file anyway? (<Enter> for yes) AT1910
  *')                                                                 AT1920
60 FORMAT (A1)                                                        AT1930
   IF (ICK(1).EQ.1.AND.ICK(2).EQ.2) THEN                              AT1940
     OPEN (UNIT=11,FILE='TEMP',STATUS='UNKNOWN')                      AT1950
     WRITE (11,70) TITL                                               AT1960
     CLOSE (11)                                                       AT1970
     GO TO 40                                                         AT1980
   END IF                                                             AT1990
70 FORMAT (' ERROR IN WELL: ',A)                                      AT2000
   CALL SET                                                           AT2010
80 CALL MODEL                                                         AT2020
   IF (ITER.EQ.25) GO TO 90                                           AT2030
   IF (RBIT.EQ.1) GO TO 80                                            AT2040
   IF (ITER.LT.2) GO TO 80                                            AT2050
   CALL PRINT                                                         AT2060
   IF (PRT(4).NE.0.AND.PRT(4).NE.9) GO TO 40                          AT2070
   CALL SAT                                                           AT2080
   GO TO 40                                                           AT2090
90 WRITE (6,160)                                                      AT2100
   WRITE (*,70) TITL                                                  AT2110
   OPEN (UNIT=11,FILE='TEMP',STATUS='UNKNOWN')                        AT2120
   WRITE (11,70) TITL                                                 AT2130
   CLOSE (11)                                                         AT2140
   GO TO 40                                                           AT2150
```

```
 100 CLOSE (UNIT=9)                                                      AT2160
      ENDFILE (UNIT=6)                                                   AT2170
      CLOSE (UNIT=5,STATUS='DELETE')                                     AT2180
      OPEN (UNIT=9,FILE='NETPATH.FIL')                                   AT2190
 110 READ (9,150,ERR=120,END=120) TITL                                   AT2200
      IF (UPCS(TITL).EQ.UPCS(DFILE)) GO TO 130                           AT2210
      GO TO 110                                                          AT2220
 120 BACKSPACE (9)                                                       AT2230
      WRITE (9,'(A)') DFILE(1:LENS(DFILE))                               AT2240
 130 CLOSE (9)                                                           AT2250
      WRITE (*,140) DFILE(1:LENS(DFILE)),DFILE(1:LENS(DFILE))            AT2260
 140 FORMAT (/,1X,A,'.PAT created.',/,1X,A,'.OUT contains the output',' AT2270
     * from WATEQF.',/)                                                  AT2280
      RETURN                                                             AT2290
C                                                                        AT2300
C                                                                        AT2310
 150 FORMAT (A80)                                                        AT2320
 160 FORMAT (4X,'CONVERGENCE DID NOT OCCUR WITHIN 25  ITERATIONS,'/4X,'  AT2330
     *CALCULATION TERMINATED',///)                                       AT2340
      END                                                                AT2350
      SUBROUTINE PREP                                                    AU  10
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                AU  20
      INTEGER D,E,DD,RBIT,CORALK,Z(120),FLAG,PRT(4),SIGN(2),PECALC,PECK  AU  30
      INTEGER ICK(2)                                                     AU  40
      DIMENSION INT(6), VAL(6), INPT(22), GRAMS(120), IEQU(50), COEF(6,  AU  50
     *200), V(120), IDH(50), IKTT(50)                                    AU  60
      DOUBLE PRECISION MI(120),KT(200),LOGKT(200),LOGKTO(200),MNTOT,     AU  70
     *LH2O,MU,NATOT,KTOT,MGTOT,LITOT,NH4TOT,KW                           AU  80
      CHARACTER*8 NSPEC(120),NREACT(200),WORD,CARD(6)                    AU  90
      CHARACTER*80 TITL,PATHLINE,PL2                                     AU 100
      COMMON MI,KT,LOGKT,LOGKTO,KW,D,E,DD,C,R,T,F,TEMP,A,B,PE,PES,PEDO,  AU 110
     *PESATO,PECK,PECALC,PH,TENMPE,TENPH,ALFA(120),GAMMA(120),AP(200),   AU 120
     *XLALFA(120),Z,CUNITS(120),ANALMI(120),GFW(120),DHA(120),NSPEC,     AU 130
     *NREACT,DH(200),AH2O,LH2O,EROR1,EROR2,EROR3,EROR4,EROR5,EHM,DENS,   AU 140
     *DOX,XLMI(120),ITER,RBIT,C1SAVE,CORALK,MU,LCHEK(200),CO2TIT,ANALCO, AU 150
     *SITOT,CATOT,MGTOT,KTOT,NATOT,SO4TOT,FETOT,PTOT,ALTOT,FTOT,BTOT,    AU 160
     *LITOT,NH4TOT,SRTOT,BATOT,CLTOT,MNTOT,ICK,PRT,TITL,EPMCAT,EPMAN,    AU 170
     *NEQU,ISPEC,KSPEC(120),IMIN,KMIN(200),TDS,IDAVES,IPRT,JJ,JK,        AU 180
     *PATHLINE,PL2,IMBAL                                                 AU 190
      DATA CARD/'CONC','EROR','DELH','TABL','LOGK','    '/,SIGN/' ','*'/ AU 200
      DATA IEQU/13,14,15,19,22,25,26,27,36,69,73,74,75,78,79,90,92,143,  AU 210
     *144,149,153,170,28*0/                                              AU 220
      DATA COEF/72*0.0,-171.9065,-0.077993,2839.319,2*0.0,71.595,6.368,- AU 230
     *0.016346,-3405.9,3*0.0,39.478,-0.065927,-12355.1,21*0.0,82.0904,0. AU 240
     *0,-3853.936,2*0.0,-29.81148,12*0.0,-171.9773,-.077993,2903.293,2*. AU 250
     *0,71.595,12*.0,.684,.0051295,4*.0,28.6059,0.012078,1573.21,2*0.0,- AU 260
     *13.2258,0.6322,-0.001225,-2835.76,51*0.0,356.3094,0.06091964,-     AU 270
     *21834.37,0.0,1684915.0,-126.8339,192*0.0,107.8871,0.03252849,-     AU 280
     *5151.79,0.0,563713.9,-38.92561,18*0.0,3.106,0.0,-673.6,3*0.0,0.    AU 290
     *991,0.00667,4*0.0,2.319,-0.011056,0.0,0.000022981,14*0.0,1209.12,. AU 300
     *31294,-34765.05,2*0.0,-478.782,-1228.732,-0.29944,35512.75,2*0.0,  AU 310
     *485.818,60*0.0,-5.3505,0.0183412,557.2461,9*0.0,11.17,-0.02386,-   AU 320
     *3279.0,303*0.0,155.0305,0.0,-7239.594,2*0.0,-56.58638,73.415,0.0,- AU 330
     *3603.341,2*0.0,-27.4437,24*0.0,-3.248,0.014867,22*0.0,-606.522,-0. AU 340
     *097611,31286.0,0.0,-2170870.0,218.68434,96*0.0,-1.019,0.012826,    AU 350
     *184*0.0/                                                           AU 360
      DATA INPT/1,2,3,4,5,6,7,35,8,45,88,62,17,18,39,51,81,85,87,90,98,  AU 370
     *101/                                                               AU 380
      C=2.302585092                                                      AU 390
      F=23.0603                                                          AU 400
```

```
      R=1.98719E-03
      EROR1=.001
      EROR2=.001
      EROR3=.001
      EROR4=.001
      EROR5=.001
      ICK(1)=0
      PEDO=100.0
      PESATO=100.
      PES=100.0
      DO 10 I=1,D
        CUNITS(I)=0.0
        ALFA(I)=0.0
        MI(I)=0.0
        XLMI(I)=0.0
        IF (Z(I).EQ.0) V(I)=1.0
        IF (Z(I).EQ.0) GO TO 10
        IF (Z(I).LT.0) V(I)=-1.0*Z(I)
        IF (Z(I).GT.0) V(I)=1.0*Z(I)
10    CONTINUE
      PECK=0
      WRITE (6,620)
      READ (5,*) TEMP,PH,EHM,EHMC,EMFZ,DENS,DOX,FLAG,CORALK,PECALC,IGO,
     *(PRT(I),I=1,4),IDAVES,ISPEC,IMIN
      IFLAG=FLAG
      IF (IPRT.EQ.1) PRT(1)=1
      IF (PRT(1).NE.0) GO TO 70
      WRITE (6,630)
      DO 30 I=1,D
        ISIG=SIGN(1)
        DO 20 J=1,NEQU
          IF (I.EQ.IEQU(J)) ISIG=SIGN(2)
20      CONTINUE
        WRITE (6,640) I,NREACT(I),DH(I),LOGKTO(I),ISIG,I,NSPEC(I),Z(I),
     *   DHA(I),GFW(I)
30    CONTINUE
      DD=D+1
      DO 50 I=DD,E
        ISIG=SIGN(1)
        DO 40 J=1,NEQU
          IF (I.EQ.IEQU(J)) ISIG=SIGN(2)
40      CONTINUE
        WRITE (6,650) I,NREACT(I),DH(I),LOGKTO(I),ISIG
50    CONTINUE
      WRITE (6,580)
      DO 60 I=1,NEQU
        WRITE (6,590) IEQU(I),NREACT(IEQU(I)),COEF(1,IEQU(I)),COEF(2,
     *   IEQU(I)),COEF(3,IEQU(I)),COEF(4,IEQU(I)),COEF(5,IEQU(I)),
     *   COEF(6,IEQU(I))
60    CONTINUE
70    IPRT=1
      WRITE (6,660) TITL
      READ (5,*) (CUNITS(INPT(I)),I=1,6)
      READ (5,*) (CUNITS(INPT(I)),I=7,12)
      IF (ISPEC.GT.0) READ (5,*) (KSPEC(I),I=1,ISPEC)
      IF (IMIN.GT.0) READ (5,*) (KMIN(I),I=1,IMIN)
80    CALL READ (WORD,INT,VAL)
      IF (WORD.NE.CARD(1)) GO TO 100
      DO 90 I=1,5
        IF (INT(I).EQ.0) GO TO 90
        CUNITS(INT(I))=VAL(I)
```

```
   90 CONTINUE                                                           AU1020
      GO TO 80                                                           AU1030
  100 IF (WORD.NE.CARD(2)) GO TO 110                                     AU1040
      EROR1=VAL(1)                                                       AU1050
      EROR2=VAL(2)                                                       AU1060
      EROR3=VAL(3)                                                       AU1070
      EROR4=VAL(4)                                                       AU1080
      EROR5=VAL(5)                                                       AU1090
      CALL READ (WORD,INT,VAL)                                           AU1100
      GO TO 100                                                          AU1110
  110 IF (WORD.NE.CARD(3)) GO TO 130                                     AU1120
      DO 120 I=1,5                                                       AU1130
        IF (INT(I).EQ.0) GO TO 120                                       AU1140
        DH(INT(I))=VAL(I)                                                AU1150
        JJ=JJ+1                                                          AU1160
        IDH(JJ)=INT(I)                                                   AU1170
        WRITE (6,670) INT(I),NREACT(INT(I)),VAL(I)                       AU1180
  120 CONTINUE                                                           AU1190
      CALL READ (WORD,INT,VAL)                                           AU1200
      GO TO 110                                                          AU1210
  130 IF (WORD.NE.CARD(4)) GO TO 150                                     AU1220
      DO 140 I=1,5                                                       AU1230
        IF (INT(I).EQ.0) GO TO 140                                       AU1240
        LOGKTO(INT(I))=VAL(I)                                            AU1250
        JK=JK+1                                                          AU1260
        IKTT(JK)=INT(I)                                                  AU1270
        WRITE (6,680) INT(I),NREACT(INT(I)),VAL(I)                       AU1280
  140 CONTINUE                                                           AU1290
      CALL READ (WORD,INT,VAL)                                           AU1300
      GO TO 130                                                          AU1310
C                                                                        AU1320
C                                                                        AU1330
C     VANT HOFF EQUATION FOR EFFECT OF T ON K                            AU1340
C                                                                        AU1350
  150 T=TEMP+273.16                                                      AU1360
      C1=(298.16-T)/(298.16*T*C*R)                                       AU1370
      DO 160 I=1,E                                                       AU1380
        LOGKT(I)=LOGKTO(I)-DH(I)*C1                                      AU1390
        LCHEK(I)=0                                                       AU1400
        IF (LOGKT(I).LT.-38.0.OR.LOGKT(I).GT.38.0) LCHEK(I)=1            AU1410
        IF (LCHEK(I).EQ.1) GO TO 160                                     AU1420
        KT(I)=10.**LOGKT(I)                                              AU1430
  160 CONTINUE                                                           AU1440
      KW=KT(153)                                                         AU1450
C                                                                        AU1460
C                                                                        AU1470
C     ANALYTICAL EXPRESSIONS FOR EFFECT OF T ON K                        AU1480
  170 IF (WORD.NE.CARD(5)) GO TO 210                                     AU1490
      IF (INT(1).EQ.0) GO TO 200                                         AU1500
      DO 180 I=1,6                                                       AU1510
        COEF(I,INT(1))=VAL(I)                                            AU1520
  180 CONTINUE                                                           AU1530
      IEQ=0                                                              AU1540
      DO 190 I=1,NEQU                                                    AU1550
        IF (IEQU(I).EQ.INT(1)) IEQ=1                                     AU1560
  190 CONTINUE                                                           AU1570
      IF (IEQ.EQ.0) NEQU=NEQU+1                                          AU1580
      IF (IEQ.EQ.0) IEQU(NEQU)=INT(1)                                    AU1590
      WRITE (6,690) INT(1),NREACT(INT(1)),COEF(1,INT(1)),COEF(2,INT(1)), AU1600
     *COEF(3,INT(1)),COEF(4,INT(1)),COEF(5,INT(1)),COEF(6,INT(1)))       AU1610
  200 CALL READ (WORD,INT,VAL)                                           AU1620
```

```
      GO TO 170                                                         AU1630
  210 IF (WORD.EQ.CARD(6)) GO TO 220                                    AU1640
      WRITE (6,600)                                                     AU1650
      CALL READ (WORD,INT,VAL)                                          AU1660
      GO TO 210                                                         AU1670
  220 READ (5,'(A80)') PATHLINE                                         AU1680
      READ (5,'(A80)') PL2                                              AU1690
      DO 270 I=1,NEQU                                                   AU1700
        IF (JJ.EQ.0) GO TO 240                                          AU1710
        DO 230 I1=1,JJ                                                  AU1720
          IF (IEQU(I).EQ.IDH(I1)) GO TO 270                             AU1730
  230   CONTINUE                                                        AU1740
  240   IF (JK.EQ.0) GO TO 260                                          AU1750
        DO 250 I1=1,JK                                                  AU1760
          IF (IEQU(I).EQ.IKTT(I1)) GO TO 270                            AU1770
  250   CONTINUE                                                        AU1780
  260   LOGKT(IEQU(I))=COEF(1,IEQU(I))+COEF(2,IEQU(I))*T+COEF(3,IEQU(I)) AU1790
     *   /T+COEF(4,IEQU(I))*T*T+COEF(5,IEQU(I))/(T*T)+COEF(6,IEQU(I))*  AU1800
     *   DLOG10(T)                                                      AU1810
        IF (IEQU(I).EQ.26) LOGKT(26)=LOGKT(26)+DLOG10(KW)               AU1820
        KT(IEQU(I))=1E1**(LOGKT(IEQU(I)))                               AU1830
  270 CONTINUE                                                          AU1840
      KW=KT(153)                                                        AU1850
C                                                                       AU1860
C                                                                       AU1870
C     CALCULATION OF ANALYZED MOLALITY                                  AU1880
      IF (FLAG.NE.0) GO TO 290                                          AU1890
      DO 280 I=1,22                                                     AU1900
        CUNITS(INPT(I))=CUNITS(INPT(I))*GFW(INPT(I))                    AU1910
  280 CONTINUE                                                          AU1920
      FLAG=2                                                            AU1930
      GO TO 310                                                         AU1940
  290 IF (FLAG.NE.1) GO TO 310                                          AU1950
      DO 300 I=1,22                                                     AU1960
        CUNITS(INPT(I))=CUNITS(INPT(I))*GFW(INPT(I))/V(INPT(I))         AU1970
  300 CONTINUE                                                          AU1980
      FLAG=2                                                            AU1990
  310 IF (FLAG.NE.2) GO TO 330                                          AU2000
      DO 320 I=1,22                                                     AU2010
        CUNITS(INPT(I))=CUNITS(INPT(I))/DENS                            AU2020
  320 CONTINUE                                                          AU2030
      FLAG=3                                                            AU2040
  330 IF (FLAG.NE.3) GO TO 360                                          AU2050
      C1=0.0                                                            AU2060
      DO 340 I=1,22                                                     AU2070
        C1=C1+CUNITS(INPT(I))                                           AU2080
  340 CONTINUE                                                          AU2090
      C1SAVE=C1                                                         AU2100
      C1=1.0/(1.0-1.0E-06*C1SAVE)                                       AU2110
      DO 350 I=1,22                                                     AU2120
        MI(INPT(I))=(CUNITS(INPT(I))/(1.0E+03*GFW(INPT(I))))*C1         AU2130
        IF (MI(INPT(I)).GT.0.0) XLMI(INPT(I))=DLOG10(MI(INPT(I)))       AU2140
        GRAMS(INPT(I))=CUNITS(INPT(I))*DENS                             AU2150
  350 CONTINUE                                                          AU2160
      C1=1.0/C1                                                         AU2170
      GO TO 400                                                         AU2180
  360 C1=0.0                                                            AU2190
      IF (FLAG.NE.4) GO TO 540                                          AU2200
      DO 380 J=1,3                                                      AU2210
        C2=0.0                                                          AU2220
        C1=1.0-C1*1.0E-06                                               AU2230
```

```
        DO 370 I=1,22
          MI(INPT(I))=CUNITS(INPT(I))/1000.
          C2=C2+MI(INPT(I))*GFW(INPT(I))*1000.*C1
  370   CONTINUE
        C1=C2
  380 CONTINUE
      C1SAVE=C1
      C1=(1.0-C1SAVE*1.0E-06)
      DO 390 I=1,22
        GRAMS(INPT(I))=MI(INPT(I))*1000.*GFW(INPT(I))*C1
        IF (MI(INPT(I)).GT.0.0) XLMI(INPT(I))=DLOG10(MI(INPT(I)))
  390 CONTINUE
  400 TDS=0.0
      DO 410 I=1,22
        ANALMI(INPT(I))=MI(INPT(I))
        TDS=TDS+GRAMS(INPT(I))
  410 CONTINUE
      EPMCAT=0.0
      EPMAN=0.0
C
C
C     CALCULATION OF CATION-ANION BALANCE
      DO 430 I=1,22
        IF (Z(INPT(I)).GT.0) GO TO 420
        EPMAN=EPMAN-Z(INPT(I))*MI(INPT(I))*C1
        GO TO 430
  420   EPMCAT=EPMCAT+Z(INPT(I))*MI(INPT(I))*C1
  430 CONTINUE
      EPMCAT=EPMCAT*1000.
      EPMAN=EPMAN*1000.
C
C
C     CALCULATION OF EH FROM FIELD DATA
      IF (EHM.LT.9.0) GO TO 470
      IF (EMFZ.GT.9.0) GO TO 440
      C1=0.429+2.4E-03*(25.0-TEMP)-EMFZ
      GO TO 450
  440 C1=0.244+8.6E-04*(25.0-TEMP)
  450 IF (EHMC.LT.9.0) GO TO 460
      GO TO 470
  460 EHM=EHMC+C1
  470 PEEH=EHM/(C*R*T/F)
      IF (PECALC.NE.0) GO TO 480
      PE=100.
      MI(8)=0.0
      MI(101)=0.0
      WRITE (6,570)
  480 IF (EHM.GE.9.0) PEEH=100.
      WRITE (6,620)
      WRITE (6,700)
      WRITE (6,710) TEMP,PH,EPMCAT,EPMAN
      WRITE (6,720) DOX,EHMC,EMFZ,IFLAG,CORALK,PECALC,IDAVES,EHM,PEEH
      IF (PECALC.EQ.1) PE=PEEH
      WRITE (6,620)
      WRITE (6,730)
      DO 490 I=1,22
        IF (MI(INPT(I)).LE.0.0) GO TO 490
        WRITE (6,760) NSPEC(INPT(I)),Z(INPT(I)),MI(INPT(I)),XLMI(INPT(I)
     *    ),GRAMS(INPT(I))
```

```
  490 CONTINUE                                                          AU2830
      WRITE (6,620)                                                     AU2840
      WRITE (6,620)                                                     AU2850
      IF (PRT(2).NE.0) GO TO 500                                        AU2860
      WRITE (6,620)                                                     AU2870
      WRITE (6,740)                                                     AU2880
  500 IF (IGO.EQ.1) GO TO 510                                           AU2890
      IF (PH.LT.3.0.OR.PH.GT.11.0) GO TO 550                            AU2900
      DUM=((EPMCAT-EPMAN)/(1.+EPMCAT+EPMAN))*100.                       AU2910
      IF (ABS(DUM).GT.30.) GO TO 550                                    AU2920
C                                                                       AU2930
C                                                                       AU2940
C     TEMPERATURE EFFECTS ON DEBYE-HUCKEL SOLVENT CONSTANTS             AU2950
  510 S1=374.11-TEMP                                                    AU2960
      S2=S1**0.333333                                                   AU2970
      S3=SQRT((1.0+0.1342489*S2-3.946263E-03*S1)/(3.1975E0-.3151548E0*  AU2980
     *S2-1.203374E-3*S1+7.48908E-13*S1**4))                             AU2990
      IF (T.LT.373.16) GO TO 520                                        AU3000
      C1=5321E0/T+233.76E0-T*(T*(8.292E-7*T-1.417E-3)+.9297E0)          AU3010
      GO TO 530                                                         AU3020
  520 C1=87.74E0-TEMP*(TEMP*(1.41E-6*TEMP-9.398E-4)+.4008E0)             AU3030
  530 C1=SQRT(C1*T)                                                     AU3040
      A=18246.0E02*S3/C1**3                                             AU3050
      B=50.29*S3/C1                                                     AU3060
      GO TO 560                                                         AU3070
  540 WRITE (6,750)                                                     AU3080
      ICK(1)=1                                                          AU3090
      GO TO 560                                                         AU3100
  550 WRITE (6,610)                                                     AU3110
      ICK(1)=1                                                          AU3120
  560 RETURN                                                            AU3130
C                                                                       AU3140
C                                                                       AU3150
  570 FORMAT (5X,'IRON AND/OR MANGANESE HAVE BEEN SPECIFIED WITHOUT REDO AU3160
     *X',/5X,'INFORMATION, IRON AND MANGANESE TOTALS HAVE BEEN CHANGED', AU3170
     *' TO ZERO.',/)                                                    AU3180
  580 FORMAT (//,1X,'"*" DENOTES THAT AN ANALYTICAL EXPRESSION FOR KT HA AU3190
     *S BEEN USED',////,1X,'SUMMARY OF ANALYTICAL EXPRESSIONS OF THE FOR AU3200
     *M',/1X,'  LOG K = A+B*T+C/T+D*T**2+E/T**2+F*LOG T'///,2X,'I   NREA AU3210
     *CT',9X,'A             B            C          D             E     AU3220
     *      F'/)                                                        AU3230
  590 FORMAT (1X,I3,2X,A8,1X,F11.4,1X,F14.9,1X,F11.4,1X,1PE11.4,1X,     AU3240
     *1PE13.6,1X,1PE14.7)                                               AU3250
  600 FORMAT (/,2X,'WARNING--- INPUT ERROR, SEARCHING FOR BLANK CARD')  AU3260
  610 FORMAT (/,2X,'WARNING---CHECK INPUT PH AND/OR CATION-ANION BALANCE AU3270
     *',/12X,'...CALCULATION TERMINATED')                               AU3280
  620 FORMAT (//)                                                       AU3290
  630 FORMAT (//,37X,'----',/,37X,'DATA',/,37X,'----',//,3X,'I',1X,'NREA AU3300
     *CT',8X,'DH',7X,'LOGKTO',10X,'I',1X,'NSPEC',6X,'Z',2X,'DHA',5X,'GFW AU3310
     *',/)                                                              AU3320
  640 FORMAT (1X,I3,1X,A8,2(1X,F10.4),A1,6X,I3,1X,A8,2X,I2,2X,F3.1,1X,  AU3330
     *F10.4)                                                            AU3340
  650 FORMAT (' ',I3,1X,A8,2(1X,F10.4),A1)                              AU3350
  660 FORMAT ('1',(A80),//)                                             AU3360
  670 FORMAT (1X,'NEW DATA *** DELTA H FOR REACTION ',I3,1X,A8,' HAS BEE AU3370
     *N CHANGED TO ',F9.4)                                              AU3380
  680 FORMAT (1X,'NEW DATA *** LOGKTO  FOR REACTION ',I3,1X,A8,' HAS BEE AU3390
     *N CHANGED TO ',F9.4)                                              AU3400
  690 FORMAT (1X,'NEW DATA *** LOGKT   FOR REACTION ',I3,1X,A8,' = ',/  AU3410
     *1X,1PE15.8,'+',E15.8,'*T+',E15.8,'/T+',E15.8,'*T**2+',E15.8,'*T**2 AU3420
     **+',E15.8,'*LOG T')                                               AU3430
```

```
  700 FORMAT (30X,'----------------',/,30X,'INITIAL SOLUTION',/,30X,'--- AU3440
     *-------------',///)                                                AU3450
  710 FORMAT (10X,'TEMPERATURE = ',F6.2,' DEGREES C    PH = ',F6.3,/,    AU3460
     *10X,'ANALYTICAL EPMCAT = ',F8.3,'    ANALYTICAL EPMAN = ',F8.3,//) AU3470
  720 FORMAT (5X,'***** OXIDATION - REDUCTION *****',///,1X,'DISSOLVED O AU3480
     *XYGEN = ',F6.3,' MG/L',/,1X,'EH MEASURED WITH CALOMEL = ',F7.4,' V AU3490
     *OLTS',9X,'FLAG CORALK PECALC IDAVES',/,1X,'MEASURED EH OF ZOBE','L AU3500
     *L SOLUTION = ',F7.4,' VOLTS',5X,I1,4X,I1,6X,I1,6X,I1,/,1X,'CORRECT AU3510
     *ED EH = ',F7.4,' VOLTS',/,1X,'PE COMPUTED FROM CORRECTED EH = ',   AU3520
     *F7.3,/)                                                            AU3530
  730 FORMAT (15X,'*** TOTAL CONCENTRATIONS OF INPUT SPECIES ***',//,    AU3540
     *25X,'TOTAL',13X,'LOG TOTAL',12X,'TOTAL',/,8X,'SPECIES',8X,'MOLALIT AU3550
     *Y',12X,'MOLALITY',11X,'MG/LITRE',/,8X,'-------',8X,'--------',12X, AU3560
     *'--------',11X,'--------',/)                                       AU3570
  740 FORMAT (22X,'*** CONVERGENCE ITERATIONS ***',//,1X,'ITERATION',2X, AU3580
     *'S1-ANALCO3',3X,'S2-SO4TOT',4X,'S3-FTOT',5X,'S4-PTOT',4X,'S5-CLTOT AU3590
     *',/)                                                               AU3600
  750 FORMAT (1X,'INPUT ERROR---UNITS OF CONCENTRATION ARE NOT KNOWN',// AU3610
     */)                                                                 AU3620
  760 FORMAT (' ',7X,A8,I3,3X,1PE12.5,9X,0PF9.4,8X,1PE12.5)              AU3630
      END                                                                AU3640
      SUBROUTINE SET                                                     AV  10
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                AV  20
      INTEGER D,E,DD,RBIT,CORALK,Z(120),PRT(4),PECALC,PECK,ICK(2)        AV  30
      DOUBLE PRECISION MI(120),KT(200),LOGKT(200),LOGKTO(200),MNTOT,     AV  40
     *LH2O,MU,NATOT,KTOT,MGTOT,LITOT,NH4TOT,KW                           AV  50
      CHARACTER*8 NSPEC(120),NREACT(200)                                 AV  60
      CHARACTER*80 TITL,PATHLINE,PL2                                     AV  70
      COMMON MI,KT,LOGKT,LOGKTO,KW,D,E,DD,C,R,T,F,TEMP,A,B,PE,PES,PEDO,  AV  80
     *PESATO,PECK,PECALC,PH,TENMPE,TENPH,ALFA(120),GAMMA(120),AP(200),   AV  90
     *XLALFA(120),Z,CUNITS(120),ANALMI(120),GFW(120),DHA(120),NSPEC,     AV 100
     *NREACT,DH(200),AH2O,LH2O,EROR1,EROR2,EROR3,EROR4,EROR5,EHM,DENS,   AV 110
     *DOX,XLMI(120),ITER,RBIT,C1SAVE,CORALK,MU,LCHEK(200),CO2TIT,ANALCO, AV 120
     *SITOT,CATOT,MGTOT,KTOT,NATOT,SO4TOT,FETOT,PTOT,ALTOT,FTOT,BTOT,    AV 130
     *LITOT,NH4TOT,SRTOT,BATOT,CLTOT,MNTOT,ICK,PRT,TITL,EPMCAT,EPMAN,    AV 140
     *NEQU,ISPEC,KSPEC(120),IMIN,KMIN(200),TDS,IDAVES,IPRT,JJ,JK,        AV 150
     *PATHLINE,PL2,IMBAL                                                 AV 160
C                                                                        AV 170
C                                                                        AV 180
C     INITIALIZE STARTING VALUES FOR ITERATIVE LOOP                      AV 190
      AH2O=1.0                                                           AV 200
      DO 10 I=1,D                                                        AV 210
        GAMMA(I)=1.0                                                     AV 220
   10 CONTINUE                                                           AV 230
      CO2TIT=MI(7)+2.0*MI(18)                                            AV 240
      ANALCO=CO2TIT                                                      AV 250
      IF (CORALK.EQ.2) CO2TIT=MI(7)+MI(18)+MI(86)                        AV 260
      SITOT=MI(35)                                                       AV 270
      CATOT=MI(1)                                                        AV 280
      MGTOT=MI(2)                                                        AV 290
      NATOT=MI(3)                                                        AV 300
      KTOT=MI(4)                                                         AV 310
      SO4TOT=MI(6)                                                       AV 320
      FETOT=MI(8)                                                        AV 330
      PTOT=MI(45)                                                        AV 340
      PIONIC=PTOT                                                        AV 350
      ALTOT=MI(51)                                                       AV 360
      FTOT=MI(62)                                                        AV 370
      BTOT=MI(87)                                                        AV 380
      LITOT=MI(81)                                                       AV 390
      NH4TOT=MI(39)                                                      AV 400
```

```
      SRTOT=MI(88)
      BATOT=MI(90)
      CLTOT=MI(5)
      MNTOT=MI(101)
      MI(35)=0.0
      MI(87)=0.0
      TENPH=10.**PH
      ALFA(64)=10.**(-PH)
C
C
C     CALCULATION OF ANION ACTIVITIES EXCEPT CO2 AND PO4 SPECIES
      ALFA(5)=MI(5)*GAMMA(5)
      ALFA(6)=MI(6)*GAMMA(6)
      ALFA(62)=MI(62)*GAMMA(62)
      ALFA(85)=MI(85)*GAMMA(85)
      ALFA(98)=MI(98)*GAMMA(98)
      ALFA(27)=AH2O*KW*TENPH
      MI(27)=ALFA(27)/GAMMA(27)
      MI(64)=1E0/(TENPH*GAMMA(64))
      ALFA(63)=ALFA(6)*KT(90)/TENPH
      MI(63)=ALFA(63)/GAMMA(63)
C
C
C     CO2 SPECIES
      IF (CORALK.EQ.2) GO TO 20
      C1=2.0*TENPH/(GAMMA(18)*KT(69))
      MI(7)=CO2TIT/(1.+GAMMA(7)*C1)
      C2=KT(36)/(TENPH*GAMMA(86))
      ALFA(7)=MI(7)*GAMMA(7)
      MI(18)=C1*ALFA(7)/2.
      MI(86)=C2*ALFA(7)
      ALFA(18)=MI(18)*GAMMA(18)
      ALFA(86)=MI(86)*GAMMA(86)
      GO TO 30
   20 MI(7)=CO2TIT/(1.0+GAMMA(7)*((KT(36)/(TENPH*GAMMA(86)))+TENPH/
     *(KT(69)*GAMMA(18))))
      MI(18)=MI(7)*GAMMA(7)*TENPH/(GAMMA(18)*KT(69))
      MI(86)=MI(7)*GAMMA(7)*KT(36)/(TENPH*GAMMA(86))
      ALFA(7)=MI(7)*GAMMA(7)
      ALFA(18)=MI(18)*GAMMA(18)
      ALFA(86)=MI(86)*GAMMA(86)
C
C
C     PHOSPHATE SPECIES
   30 MI(45)=PTOT/(1.+(KT(17)*GAMMA(45)/(GAMMA(48)*TENPH**2))+(KT(16)*
     *GAMMA(45)/(TENPH*GAMMA(47))))
      ALFA(45)=MI(45)*GAMMA(45)
      ALFA(47)=KT(16)*ALFA(45)/TENPH
      MI(47)=ALFA(47)/GAMMA(47)
      ALFA(48)=KT(17)*ALFA(45)/(TENPH**2)
      MI(48)=ALFA(48)/GAMMA(48)
      ITER=0
      RETURN
      END
      SUBROUTINE READ (WORD,INT,VAL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*8 WORD
      DIMENSION INT(6), VAL(6)
      CHARACTER*80 LINE
      READ (5,10) LINE
   10 FORMAT (A80)
```

```
      OPEN (UNIT=13,FILE='WATEQFP.SCR')
      WRITE (13,10) LINE
      REWIND (13)
      READ (13,*) WORD,(INT(I),VAL(I),I=1,5)
      IF (WORD.NE.'LOGK') THEN
        CLOSE (13,STATUS='DELETE')
        RETURN
      END IF
      REWIND (13)
      READ (13,*) WORD,(INT(I),VAL(I),I=1,6)
      CLOSE (13,STATUS='DELETE')
      RETURN
      END
      SUBROUTINE MODEL
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER D,E,DD,RBIT,CORALK,Z(120),LIST(8),LIST1(5),LIST2(18),
     *LIST3(9),PRT(4),PECALC,PECK,ICK(2)
      DOUBLE PRECISION MI(120),KT(200),LOGKT(200),LOGKTO(200),MNTOT,
     *LH2O,MU,NATOT,KTOT,MGTOT,LITOT,NH4TOT,KW,MUHALF,L1ALK(11)
      CHARACTER*8 NSPEC(120),NREACT(200)
      CHARACTER*80 TITL,PATHLINE,PL2
      DIMENSION NPAIR(5), L1M(11), L1K(11), L1C(11), L1A(11), L2M(14),
     *L2K(14), L2C(14), L3M(7), L3K(7), L3C(7), L4M(14), L4K(14),
     *L4C(14), L4A(14), L5M(9), L5K(9), L5C(9)
      COMMON MI,KT,LOGKT,LOGKTO,KW,D,E,DD,C,R,T,F,TEMP,A,B,PE,PES,PEDO,
     *PESATO,PECK,PECALC,PH,TENMPE,TENPH,ALFA(120),GAMMA(120),AP(200),
     *XLALFA(120),Z,CUNITS(120),ANALMI(120),GFW(120),DHA(120),NSPEC,
     *NREACT,DH(200),AH2O,LH2O,EROR1,EROR2,EROR3,EROR4,EROR5,EHM,DENS,
     *DOX,XLMI(120),ITER,RBIT,C1SAVE,CORALK,MU,LCHEK(200),CO2TIT,ANALCO,
     *SITOT,CATOT,MGTOT,KTOT,NATOT,SO4TOT,FETOT,PTOT,ALTOT,FTOT,BTOT,
     *LITOT,NH4TOT,SRTOT,BATOT,CLTOT,MNTOT,ICK,PRT,TITL,EPMCAT,EPMAN,
     *NEQU,ISPEC,KSPEC(120),IMIN,KMIN(200),TDS,IDAVES,IPRT,JJ,JK,
     *PATHLINE,PL2,IMBAL
      DATA LIST/17,35,66,70,71,72,84,87/
      DATA LIST1/42,43,44,50,94/
      DATA LIST2/8,9,10,11,12,13,15,16,28,33,34,65,77,78,79,80,100,99/
      DATA LIST3/82,83,88,89,69,97,90,91,114/
      DATA L1M/7,21,22,30,31,42,43,86,111,69,97/,L1K/69,74,75,78,79,70,
     *71,36,167,149,170/,L1C/64,2,2,1,1,3,3,64,101,88,88/,L1A/18,18,7,7,
     *18,18,7,7,7,7,18/,L1ALK/1.0,2.0,1.0,1.0,2.0,2.0,1.0,0.0,1.0,1.0,2.
     *0/,L2M/15,23,32,34,44,46,59,60,63,83,92,96,109,114/,L2K/5,76,24,9,
     *72,73,88,89,90,127,132,136,165,182/,L2C/8,2,1,8,3,4,51,51,64,81,
     *39,64,101,88/,L3M/20,55,56,57,58,108,49/,L3K/23,84,85,86,87,164,
     *80/,L3C/2,51,51,51,51,101,1/,L4M/13,40,41,47,48,50,61,65,73,74,75,
     *76,99,100/,L4K/140,124,125,16,17,31,33,121,34,35,122,123,157,139/,
     *L4C/8,2,2,64,64,3,4,8,2,1,1,1,8,8/,L4A/47,45,48,45,45,47,47,48,47,
     *47,45,48,48,47/,L5M/16,28,33,93,94,95,103,104,105/,L5K/6,7,8,133,
     *134,135,159,160,161/,L5C/8,8,8,64,3,4,101,101,101/,NPAIR/11,14,7,
     *14,9/
      ITER=ITER+1
C
C
C     CALCULATION OF TOTAL MOLALITY AND AH2O
      J=1
      C1=0.0
      DO 10 I=1,D
        IF (I.EQ.LIST(J)) THEN
          IF (J.LT.8) J=J+1
        ELSE
          C1=C1+MI(I)
        END IF
```

```
C      IF (I.EQ.LIST(J)) GO TO 10
C      C1=C1+MI(I)
C      GO TO 20
C   10 J=J+1
   10 CONTINUE
      AH2O=1.0-0.017*C1
      LH2O=DLOG10(AH2O)
      IF (DOX.GT.0.0) PEDO=-(DLOG10(KT(152))+PH+0.5*LH2O-0.25*
     *DLOG10(DOX/32E3))
      IF (DOX.GT.0.0) PESATO=-(DLOG10(KT(137))+PH+0.5*LH2O-0.25*
     *DLOG10(DOX/32E3))
      IF (PECALC.EQ.2) PE=PEDO
      IF (PECALC.EQ.3) PE=PESATO
C
C
C     CALCULATION OF ACTIVITY COEFFICIENTS
      MU=0.0
      J=1
      DO 20 I=1,D
        IF (I.EQ.LIST(J)) THEN
          IF (J.LT.8) J=J+1
        ELSE
          MU=MU+0.5*MI(I)*Z(I)*Z(I)
        END IF
C      IF (I.EQ.LIST(J)) GO TO 30
C      MU=MU+0.5*MI(I)*Z(I)*Z(I)
C      GO TO 40
C   30 J=J+1
   20 CONTINUE
      MUHALF=SQRT(MU)
      C1=-A*4E0*MUHALF
      GAMMA(1)=1E1**(C1/(1E0+B*5E0*MUHALF)+0.165*MU)
      GAMMA(2)=1E1**(C1/(1E0+B*5.5*MUHALF)+0.2*MU)
      GAMMA(3)=1E1**(-A*MUHALF/(1E0+B*4E0*MUHALF)+0.075*MU)
      GAMMA(4)=1E1**(-A*MUHALF/(1E0+B*3.5*MUHALF)+0.015*MU)
      GAMMA(5)=GAMMA(4)
      GAMMA(6)=1E1**(C1/(1E0+B*5E0*MUHALF)-0.04*MU)
      DO 40 I=8,D
        IF (Z(I).EQ.0) GO TO 30
        IF (IDAVES.EQ.1) GAMMA(I)=1E1**(-A*Z(I)**2*((MUHALF/(1.0+MUHALF)
     *   -0.3*MU)))
        IF (IDAVES.EQ.1) GO TO 40
        GAMMA(I)=1E1**(-A*MUHALF*Z(I)**2/(1E0+DHA(I)*B*MUHALF))
        GO TO 40
   30   GAMMA(I)=10.**(0.1*MU)
   40 CONTINUE
      GAMMA(7)=1E1**(-A*MUHALF*Z(7)**2/(1E0+DHA(7)*B*MUHALF))
      GAMMA(18)=1E1**(-A*MUHALF*Z(18)**2/(1E0+DHA(18)*B*MUHALF))
      GAMMA(86)=1E1**(MU*(170.01/T-.8798+.0013935*T)+MU*MU*(28.81/T-.
     *2108+.0003641*T))
      GAMMA(30)=GAMMA(7)
      GAMMA(31)=1E1**(-0.5*MU)
C
C
C     SULFUR SPECIES AND PE CALCULATION FROM S
      C1=KT(92)*TENPH/GAMMA(67)
      C2=KT(92)*KT(93)*TENPH**2/GAMMA(68)
      MI(14)=MI(17)/(1E0+GAMMA(14)*(C1+C2))
      ALFA(14)=MI(14)*GAMMA(14)
      ALFA(17)=MI(17)*GAMMA(17)
      MI(67)=ALFA(14)*C1
```

```
      MI(68)=ALFA(14)*C2                                                AX1100
      ALFA(67)=MI(67)*GAMMA(67)                                         AX1110
      ALFA(68)=MI(68)*GAMMA(68)                                         AX1120
      C1=ALFA(6)*ALFA(14)                                               AX1130
      IF (C1.GT.0.0) GO TO 50                                           AX1140
      GO TO 60                                                          AX1150
   50 PES=0.125*LOGKT(91)+0.125*DLOG10(ALFA(6))-1.25*PH-0.125*          AX1160
     *DLOG10(ALFA(14))-0.5*LH2O                                         AX1170
      IF (PECALC.EQ.4) PE=PES                                           AX1180
   60 IF (PECALC.EQ.0.0.OR.PE.GE.100.) GO TO 70                         AX1190
      TENMPE=10.**(-PE)                                                 AX1200
      GO TO 80                                                          AX1210
   70 TENMPE=1.0                                                        AX1220
C                                                                       AX1230
C                                                                       AX1240
C     SILICA SPECIES                                                    AX1250
   80 C1=KT(14)*TENPH/GAMMA(25)                                         AX1260
      C2=KT(15)*TENPH**2/GAMMA(26)                                      AX1270
      MI(24)=SITOT/(1.0+GAMMA(24)*(C1+C2))                              AX1280
      ALFA(24)=MI(24)*GAMMA(24)                                         AX1290
      MI(25)=ALFA(24)*C1                                                AX1300
      MI(26)=ALFA(24)*C2                                                AX1310
      ALFA(25)=MI(25)*GAMMA(25)                                         AX1320
      ALFA(26)=MI(26)*GAMMA(26)                                         AX1330
C                                                                       AX1340
C                                                                       AX1350
C     BORON SPECIES                                                     AX1360
      C1=GAMMA(36)*KT(26)*TENPH/GAMMA(37)                               AX1370
      MI(36)=BTOT/(1.0+C1)                                              AX1380
      MI(37)=C1*MI(36)                                                  AX1390
      ALFA(36)=MI(36)*GAMMA(36)                                         AX1400
      ALFA(37)=MI(37)*GAMMA(37)                                         AX1410
C                                                                       AX1420
C                                                                       AX1430
C     NITROGEN SPECIES                                                  AX1440
      C1=TENPH*KT(27)/GAMMA(38)                                         AX1450
      C2=ALFA(6)*KT(132)/GAMMA(92)                                      AX1460
      MI(39)=NH4TOT/(1E0+GAMMA(39)*(C1+C2))                             AX1470
      ALFA(39)=MI(39)*GAMMA(39)                                         AX1480
      MI(38)=ALFA(39)*C1                                                AX1490
      ALFA(38)=MI(38)*GAMMA(38)                                         AX1500
      MI(92)=ALFA(39)*C2                                                AX1510
      ALFA(92)=MI(92)*GAMMA(92)                                         AX1520
C                                                                       AX1530
C                                                                       AX1540
C     MAGNESIUM SPECIES                                                 AX1550
      MI(19)=ALFA(27)*KT(25)/GAMMA(19)                                  AX1560
      MI(20)=ALFA(62)*KT(23)/GAMMA(20)                                  AX1570
      MI(21)=ALFA(18)*KT(74)/GAMMA(21)                                  AX1580
      MI(22)=ALFA(7)*KT(75)/GAMMA(22)                                   AX1590
      MI(23)=ALFA(6)*KT(76)/GAMMA(23)                                   AX1600
      MI(40)=ALFA(45)*KT(124)/GAMMA(40)                                 AX1610
      MI(41)=ALFA(48)*KT(125)/GAMMA(41)                                 AX1620
      MI(73)=ALFA(47)*KT(34)/GAMMA(73)                                  AX1630
      MI(2)=MGTOT/(1.0+GAMMA(2)*(MI(19)+MI(20)+MI(21)+MI(22)+MI(23)+    AX1640
     *MI(40)+MI(41)+MI(73)))                                            AX1650
      ALFA(2)=MI(2)*GAMMA(2)                                            AX1660
      C1=ALFA(2)                                                        AX1670
      DO 90 I=19,23                                                     AX1680
        MI(I)=C1*MI(I)                                                  AX1690
        ALFA(I)=MI(I)*GAMMA(I)                                          AX1700
```

```
   90 CONTINUE                                                          AX1710
      MI(40)=C1*MI(40)                                                  AX1720
      ALFA(40)=MI(40)*GAMMA(40)                                         AX1730
      MI(41)=C1*MI(41)                                                  AX1740
      ALFA(41)=MI(41)*GAMMA(41)                                         AX1750
      MI(73)=C1*MI(73)                                                  AX1760
      ALFA(73)=MI(73)*GAMMA(73)                                         AX1770
C                                                                       AX1780
C                                                                       AX1790
C     CALCIUM SPECIES                                                   AX1800
      MI(29)=ALFA(27)*KT(77)/GAMMA(29)                                  AX1810
      MI(30)=ALFA(7)*KT(78)/GAMMA(30)                                   AX1820
      MI(31)=ALFA(18)*KT(79)/GAMMA(31)                                  AX1830
      MI(32)=ALFA(6)*KT(24)/GAMMA(32)                                   AX1840
      MI(74)=ALFA(47)*KT(35)/GAMMA(74)                                  AX1850
      MI(76)=ALFA(48)*KT(123)/GAMMA(76)                                 AX1860
      MI(75)=ALFA(45)*KT(122)/GAMMA(75)                                 AX1870
      MI(49)=ALFA(62)*KT(80)/GAMMA(49)                                  AX1880
      MI(1)=CATOT/(1.0+GAMMA(1)*(MI(29)+MI(30)+MI(31)+MI(32)+MI(74)+    AX1890
     *MI(75)+MI(76)+MI(49)))                                            AX1900
      C1=MI(1)*GAMMA(1)                                                 AX1910
      ALFA(1)=C1                                                        AX1920
      DO 100 I=29,32                                                    AX1930
        MI(I)=C1*MI(I)                                                  AX1940
        ALFA(I)=MI(I)*GAMMA(I)                                          AX1950
  100 CONTINUE                                                          AX1960
      MI(74)=C1*MI(74)                                                  AX1970
      ALFA(74)=MI(74)*GAMMA(74)                                         AX1980
      MI(75)=C1*MI(75)                                                  AX1990
      ALFA(75)=MI(75)*GAMMA(75)                                         AX2000
      MI(76)=C1*MI(76)                                                  AX2010
      ALFA(76)=MI(76)*GAMMA(76)                                         AX2020
      MI(49)=C1*MI(49)                                                  AX2030
      ALFA(49)=MI(49)*GAMMA(49)                                         AX2040
C                                                                       AX2050
C                                                                       AX2060
C     SODIUM SPECIES                                                    AX2070
      MI(42)=ALFA(18)*KT(70)/GAMMA(42)                                  AX2080
      MI(43)=ALFA(7)*KT(71)/GAMMA(43)                                   AX2090
      MI(44)=ALFA(6)*KT(72)/GAMMA(44)                                   AX2100
      MI(50)=ALFA(47)*KT(31)/GAMMA(50)                                  AX2110
      MI(94)=ALFA(5)*KT(134)/GAMMA(94)                                  AX2120
      MI(3)=NATOT/(1.0+GAMMA(3)*(MI(42)+MI(43)+MI(44)+MI(50)+MI(94)))   AX2130
      ALFA(3)=MI(3)*GAMMA(3)                                            AX2140
      C1=ALFA(3)                                                        AX2150
      DO 110 I=1,5                                                      AX2160
        MI(LIST1(I))=C1*MI(LIST1(I))                                    AX2170
        ALFA(LIST1(I))=MI(LIST1(I))*GAMMA(LIST1(I))                     AX2180
  110 CONTINUE                                                          AX2190
C                                                                       AX2200
C                                                                       AX2210
C     POTASSIUM SPECIES                                                 AX2220
      MI(46)=ALFA(6)*KT(73)/GAMMA(46)                                   AX2230
      MI(61)=ALFA(47)*KT(33)/GAMMA(61)                                  AX2240
      MI(95)=ALFA(5)*KT(135)/GAMMA(95)                                  AX2250
      MI(4)=KTOT/(1.0+GAMMA(4)*(MI(46)+MI(61)+MI(95)))                  AX2260
      ALFA(4)=MI(4)*GAMMA(4)                                            AX2270
      C1=ALFA(4)                                                        AX2280
      MI(46)=C1*MI(46)                                                  AX2290
      ALFA(46)=MI(46)*GAMMA(46)                                         AX2300
      MI(61)=C1*MI(61)                                                  AX2310
```

```
      ALFA(61)=MI(61)*GAMMA(61)                                         AX2320
      MI(95)=C1*MI(95)                                                  AX2330
      ALFA(95)=MI(95)*GAMMA(95)                                         AX2340
C                                                                       AX2350
C                                                                       AX2360
C     ALUMINIUM SPECIES                                                 AX2370
      MI(52)=ALFA(27)*KT(81)/GAMMA(52)                                  AX2380
      MI(53)=ALFA(27)**2*KT(82)/GAMMA(53)                               AX2390
      MI(54)=ALFA(27)**4*KT(83)/GAMMA(54)                               AX2400
      MI(55)=ALFA(62)*KT(84)/GAMMA(55)                                  AX2410
      MI(56)=ALFA(62)**2*KT(85)/GAMMA(56)                               AX2420
      MI(57)=ALFA(62)**3*KT(86)/GAMMA(57)                               AX2430
      MI(58)=ALFA(62)**4*KT(87)/GAMMA(58)                               AX2440
      MI(59)=ALFA(6)*KT(88)/GAMMA(59)                                   AX2450
      MI(60)=ALFA(6)**2*KT(89)/GAMMA(60)                                AX2460
      MI(51)=ALTOT/(1.0+GAMMA(51)*(MI(52)+MI(53)+MI(54)+MI(55)+MI(56)+  AX2470
     *MI(57)+MI(58)+MI(59)+MI(60)))                                     AX2480
      ALFA(51)=MI(51)*GAMMA(51)                                         AX2490
      C1=ALFA(51)                                                       AX2500
      DO 120 I=52,60                                                    AX2510
        MI(I)=C1*MI(I)                                                  AX2520
        ALFA(I)=MI(I)*GAMMA(I)                                          AX2530
  120 CONTINUE                                                          AX2540
C                                                                       AX2550
C                                                                       AX2560
C     IRON SPECIES                                                      AX2570
      IF (ABS(PE).LT.20.0.AND.FETOT.GT.0.0) GO TO 130                   AX2580
      GO TO 150                                                         AX2590
  130 MI(9)=KT(1)/(TENMPE*GAMMA(9))                                     AX2600
      MI(10)=KT(2)*AH2O*TENPH/(TENMPE*GAMMA(10))                        AX2610
      MI(11)=KT(3)*AH2O*TENPH/GAMMA(11)                                 AX2620
      MI(12)=KT(4)*AH2O**3*TENPH**3/GAMMA(12)                           AX2630
      MI(13)=KT(140)*ALFA(47)/(GAMMA(13)*TENMPE)                        AX2640
      MI(15)=KT(5)*ALFA(6)/(TENMPE*GAMMA(15))                           AX2650
      MI(16)=KT(6)*ALFA(5)/(TENMPE*GAMMA(16))                           AX2660
      MI(28)=KT(7)*ALFA(5)**2/(TENMPE*GAMMA(28))                        AX2670
      MI(33)=KT(8)*ALFA(5)**3/(TENMPE*GAMMA(33))                        AX2680
      MI(34)=KT(9)*ALFA(6)/GAMMA(34)                                    AX2690
      MI(65)=KT(121)*ALFA(48)/GAMMA(65)                                 AX2700
      MI(77)=KT(103)*(AH2O*TENPH)**2/(TENMPE*GAMMA(77))                 AX2710
      MI(78)=KT(104)*(AH2O*TENPH)**3/(TENMPE*GAMMA(78))                 AX2720
      MI(79)=KT(105)*(AH2O*TENPH)**4/(TENMPE*GAMMA(79))                 AX2730
      MI(80)=KT(106)*(AH2O*TENPH)**2/GAMMA(80)                          AX2740
      MI(99)=KT(157)*ALFA(48)/(TENMPE*GAMMA(99))                        AX2750
      MI(100)=KT(139)*ALFA(47)/GAMMA(100)                               AX2760
      MI(8)=FETOT/(1.0+GAMMA(8)*(MI(9)+MI(10)+MI(11)+MI(12)+MI(13)+     AX2770
     *MI(15)+MI(16)+MI(28)+MI(33)+MI(34)+MI(65)+MI(77)+MI(78)+MI(79)+   AX2780
     *MI(80)+MI(100)+MI(99)))                                           AX2790
      ALFA(8)=MI(8)*GAMMA(8)                                            AX2800
      C1=ALFA(8)                                                        AX2810
      DO 140 I=2,18                                                     AX2820
        MI(LIST2(I))=C1*MI(LIST2(I))                                    AX2830
        ALFA(LIST2(I))=MI(LIST2(I))*GAMMA(LIST2(I))                     AX2840
  140 CONTINUE                                                          AX2850
      GO TO 170                                                         AX2860
  150 DO 160 I=2,18                                                     AX2870
        MI(LIST2(I))=0.0                                                AX2880
  160 CONTINUE                                                          AX2890
      ALFA(8)=MI(8)*GAMMA(8)                                            AX2900
C                                                                       AX2910
C     MANGANESE SPECIES                                                 AX2920
```

```
C                                                                        AX2930
  170 IF (ABS(PE).LT.20.0.AND.MNTOT.GT.0.0) GO TO 180                    AX2940
      GO TO 220                                                          AX2950
  180 MI(102)=KT(158)/(GAMMA(102)*TENMPE)                                AX2960
      MI(103)=KT(159)*MI(5)*GAMMA(5)/GAMMA(103)                          AX2970
      MI(104)=KT(160)*MI(5)**2*GAMMA(5)**2/GAMMA(104)                    AX2980
      MI(105)=KT(161)*MI(5)**3*GAMMA(5)**3/GAMMA(105)                    AX2990
      MI(106)=KT(162)*MI(27)*GAMMA(27)/GAMMA(106)                        AX3000
      MI(107)=KT(163)*MI(27)**3*GAMMA(27)**3/GAMMA(107)                  AX3010
      MI(108)=KT(164)*MI(62)*GAMMA(62)/GAMMA(108)                        AX3020
      MI(109)=KT(165)*MI(6)*GAMMA(6)/GAMMA(109)                          AX3030
      MI(110)=KT(166)*MI(85)**2*GAMMA(85)**2/GAMMA(110)                  AX3040
      MI(111)=KT(167)*MI(7)*GAMMA(7)/GAMMA(111)                          AX3050
      XMI112=LOGKT(168)+4*LH2O-(DLOG10(GAMMA(112))-8*PH-5*PE)            AX3060
      IF (XMI112.LT.-38.) MI(112)=0.0                                    AX3070
      IF (XMI112.LT.-38.) GO TO 190                                      AX3080
      MI(112)=10.**XMI112                                                AX3090
  190 XMI113=LOGKT(169)+4*LH2O-(DLOG10(GAMMA(113))-8*PH-4*PE)            AX3100
      IF (XMI113.LT.-38.) MI(113)=0.0                                    AX3110
      IF (XMI113.LT.-38.) GO TO 200                                      AX3120
      MI(113)=10.**XMI113                                                AX3130
  200 MI(115)=KT(171)*AH2O**2/(GAMMA(115)*ALFA(64)**3)                   AX3140
      MI(101)=MNTOT/(1.0+GAMMA(101)*(MI(102)+MI(103)+MI(104)+MI(105)+    AX3150
     *MI(106)+MI(107)+MI(108)+MI(109)+MI(110)+MI(111)+MI(112)+MI(113)+   AX3160
     *MI(115)))                                                          AX3170
      ALFA(101)=MI(101)*GAMMA(101)                                       AX3180
      C1=ALFA(101)                                                       AX3190
      DO 210 I=102,113                                                   AX3200
        MI(I)=C1*MI(I)                                                   AX3210
        ALFA(I)=MI(I)*GAMMA(I)                                           AX3220
  210 CONTINUE                                                           AX3230
      MI(115)=C1*MI(115)                                                 AX3240
      ALFA(115)=MI(115)*GAMMA(115)                                       AX3250
      GO TO 240                                                          AX3260
  220 DO 230 I=101,113                                                   AX3270
        MI(I)=0.0                                                        AX3280
  230 CONTINUE                                                           AX3290
      MI(115)=0.0                                                        AX3300
C                                                                        AX3310
C                                                                        AX3320
C     CALCULATION OF PO2 AND PCH4                                        AX3330
  240 IF (ABS(PE).LT.19.0) GO TO 250                                     AX3340
      GO TO 260                                                          AX3350
  250 C1=DLOG10(KT(94))+PH+PE+0.5*LH2O                                   AX3360
      ALFA(70)=10.**(4.0*C1)                                             AX3370
  260 IF (ABS(PE).LT.19.0.AND.ALFA(7).GT.0.0) GO TO 270                  AX3380
      GO TO 280                                                          AX3390
  270 XLALFA(71)=(DLOG10(KT(95))-8.0*PE-9.0*PH-3.0*LH2O+DLOG10(ALFA(7))) AX3400
      IF (XLALFA(71).LT.-38.) GO TO 280                                  AX3410
      ALFA(71)=10.**XLALFA(71)                                           AX3420
C                                                                        AX3430
C                                                                        AX3440
C     LITHIUM, STRONTIUM, BARIUM SPECIES                                 AX3450
  280 C1=KT(126)*ALFA(27)/GAMMA(82)                                      AX3460
      C2=KT(127)*ALFA(6)/GAMMA(83)                                       AX3470
      MI(81)=LITOT/(1.0+GAMMA(81)*(C1+C2))                               AX3480
      ALFA(81)=MI(81)*GAMMA(81)                                          AX3490
      MI(82)=C1*ALFA(81)                                                 AX3500
      MI(83)=C2*ALFA(81)                                                 AX3510
      C1=KT(130)*ALFA(27)/GAMMA(89)                                      AX3520
      C2=KT(149)*ALFA(7)/GAMMA(69)                                       AX3530
```

```
      C3=KT(170)*ALFA(18)/GAMMA(97)
      C5=KT(182)*ALFA(6)/GAMMA(114)
      MI(88)=SRTOT/(1.0+GAMMA(88)*(C1+C2+C3+C5))
      C4=GAMMA(88)*MI(88)
      MI(89)=C1*C4
      MI(69)=C2*C4
      MI(97)=C3*C4
      MI(114)=C5*C4
      C1=KT(131)*ALFA(27)/GAMMA(91)
      MI(90)=BATOT/(1.0+GAMMA(90)*C1)
      MI(91)=GAMMA(90)*MI(90)*C1
      DO 290 I=1,9
        ALFA(LIST3(I))=MI(LIST3(I))*GAMMA(LIST3(I))
  290 CONTINUE
C     SUMMATION OF ANION SPECIES
      N=NPAIR(1)
      IF (CORALK.NE.2) GO TO 310
      S1=MI(18)
      DO 300 I=1,N
        S1=S1+MI(L1M(I))
  300 CONTINUE
      GO TO 330
  310 S1=2.0*MI(18)
      DO 320 I=1,N
        S1=S1+L1ALK(I)*MI(L1M(I))
  320 CONTINUE
  330 N=NPAIR(2)
      S2=MI(6)+MI(60)
      DO 340 I=1,N
        S2=S2+MI(L2M(I))
  340 CONTINUE
      N=NPAIR(3)
      S3=MI(62)+MI(56)+2.0*MI(57)+3.0*MI(58)
      DO 350 I=1,N
        S3=S3+MI(L3M(I))
  350 CONTINUE
      N=NPAIR(4)
      S4=MI(45)
      DO 360 I=1,N
        S4=S4+MI(L4M(I))
  360 CONTINUE
      N=NPAIR(5)
      S5=MI(5)+MI(28)+MI(104)+2.0*(MI(33)+MI(105))
      DO 370 I=1,N
        S5=S5+MI(L5M(I))
  370 CONTINUE
      ANALCO=CO2TIT
C     MASS BALANCE ON CARBON
      IF (CO2TIT.LE.0.0) GO TO 440
      ACT=KT(69)*ALFA(64)
      SUM=0.0
      SUM1=0.0
      N=NPAIR(1)
      DO 380 I=1,N
        MI(L1M(I))=KT(L1K(I))*ALFA(L1C(I))/GAMMA(L1M(I))
        IF (L1A(I).EQ.7) MI(L1M(I))=MI(L1M(I))*ACT
        SUM=SUM+MI(L1M(I))
        SUM1=SUM1+L1ALK(I)*MI(L1M(I))
  380 CONTINUE
      IF (CORALK.NE.2) GO TO 400
      MI(18)=ANALCO/(1.0+GAMMA(18)*SUM)
```

```
      ALFA(18)=MI(18)*GAMMA(18)
      DO 390 I=1,N
        MI(L1M(I))=MI(L1M(I))*ALFA(18)
        ALFA(L1M(I))=MI(L1M(I))*GAMMA(L1M(I))
  390 CONTINUE
      GO TO 440
  400 IF (CORALK.EQ.1) GO TO 420
      IF (CORALK.EQ.3) GO TO 410
      ANALCO=CO2TIT-MI(25)-2.0*MI(26)-MI(27)-MI(37)-2.0*MI(45)-MI(47)-
     *MI(54)-MI(67)-2.0*MI(68)-MI(82)
      GO TO 420
  410 SUMALK=MI(29)+MI(76)+MI(19)+MI(41)+MI(11)+MI(10)+MI(99)+MI(52)+
     *MI(106)+MI(25)+MI(37)+MI(82)+MI(89)+MI(91)+MI(38)+MI(67)+MI(48)+
     *MI(27)+2.0*(MI(74)+MI(73)+MI(50)+MI(61)+MI(80)+MI(100)+MI(13)+
     *MI(77)+MI(53)+MI(26)+MI(68)+MI(47))+3.0*(MI(75)+MI(40)+MI(12)+
     *MI(78)+MI(107)+MI(115)+MI(45))+4.0*(MI(79)+MI(54))-MI(64)-MI(63)-
     *MI(93)-2.0*MI(96)+MI(69)+2.0*MI(97)
      ANALCO=CO2TIT-SUMALK
  420 IF (ANALCO.LT.0.0) ANALCO=0.0
      MI(18)=ANALCO/(2.0+GAMMA(18)*SUM1)
      ALFA(18)=MI(18)*GAMMA(18)
      DO 430 I=1,N
        MI(L1M(I))=MI(L1M(I))*ALFA(18)
        ALFA(L1M(I))=MI(L1M(I))*GAMMA(L1M(I))
  430 CONTINUE
C     MASS BALANCE ON SULFATE
  440 IF (SO4TOT.LE.0.0) GO TO 480
      N=NPAIR(2)
      DO 450 I=1,N
        MI(L2M(I))=KT(L2K(I))*ALFA(L2C(I))/GAMMA(L2M(I))
  450 CONTINUE
      MI(15)=MI(15)/TENMPE
      MI(60)=MI(60)*ALFA(6)
      MI(96)=MI(96)*ALFA(64)
      SUM=MI(60)
      DO 460 I=1,N
        SUM=SUM+MI(L2M(I))
  460 CONTINUE
      MI(6)=SO4TOT/(1.0+GAMMA(6)*SUM)
      ALFA(6)=MI(6)*GAMMA(6)
      DO 470 I=1,N
        MI(L2M(I))=MI(L2M(I))*ALFA(6)
        ALFA(L2M(I))=MI(L2M(I))*GAMMA(L2M(I))
  470 CONTINUE
C     MASS BALANCE ON FLUORIDE
  480 IF (FTOT.LE.0.0) GO TO 520
      N=NPAIR(3)
      DO 490 I=1,N
        MI(L3M(I))=KT(L3K(I))*ALFA(L3C(I))/GAMMA(L3M(I))
  490 CONTINUE
      MI(56)=MI(56)*ALFA(62)
      MI(57)=MI(57)*ALFA(62)*ALFA(62)
      MI(58)=MI(58)*ALFA(62)*ALFA(62)*ALFA(62)
      SUM=MI(56)+2.0*MI(57)+3.0*MI(58)
      DO 500 I=1,N
        SUM=SUM+MI(L3M(I))
  500 CONTINUE
      MI(62)=FTOT/(1.0+GAMMA(62)*SUM)
      ALFA(62)=MI(62)*GAMMA(62)
      DO 510 I=1,N
        MI(L3M(I))=MI(L3M(I))*ALFA(62)
```

```
         ALFA(L3M(I))=MI(L3M(I))*GAMMA(L3M(I))                             AX4760
  510 CONTINUE                                                             AX4770
C     MASS BALANCE ON PHOSPHATE                                            AX4780
  520 IF (PTOT.LE.0.0) GO TO 560                                           AX4790
      N=NPAIR(4)                                                           AX4800
      C1=KT(16)*ALFA(64)                                                   AX4810
      C2=KT(17)*ALFA(64)*ALFA(64)                                          AX4820
      DO 530 I=1,N                                                         AX4830
        MI(L4M(I))=KT(L4K(I))*ALFA(L4C(I))/GAMMA(L4M(I))                   AX4840
        IF (L4A(I).EQ.47) MI(L4M(I))=MI(L4M(I))*C1                         AX4850
        IF (L4A(I).EQ.48) MI(L4M(I))=MI(L4M(I))*C2                         AX4860
  530 CONTINUE                                                             AX4870
      MI(13)=MI(13)/TENMPE                                                 AX4880
      MI(48)=MI(48)*ALFA(64)                                               AX4890
      MI(99)=MI(99)/TENMPE                                                 AX4900
      SUM=0.0                                                              AX4910
      DO 540 I=1,N                                                         AX4920
        SUM=SUM+MI(L4M(I))                                                 AX4930
  540 CONTINUE                                                             AX4940
      MI(45)=PTOT/(1.0+GAMMA(45)*SUM)                                      AX4950
      ALFA(45)=MI(45)*GAMMA(45)                                            AX4960
      DO 550 I=1,N                                                         AX4970
        MI(L4M(I))=MI(L4M(I))*ALFA(45)                                     AX4980
        ALFA(L4M(I))=MI(L4M(I))*GAMMA(L4M(I))                              AX4990
  550 CONTINUE                                                             AX5000
C     MASS BALANCE ON CHLORIDE                                             AX5010
  560 IF (CLTOT.LE.0.0) GO TO 600                                          AX5020
      N=NPAIR(5)                                                           AX5030
      DO 570 I=1,N                                                         AX5040
        MI(L5M(I))=KT(L5K(I))*ALFA(L5C(I))/GAMMA(L5M(I))                   AX5050
  570 CONTINUE                                                             AX5060
      MI(16)=MI(16)/TENMPE                                                 AX5070
      MI(28)=MI(28)*ALFA(5)/TENMPE                                         AX5080
      MI(33)=MI(33)*ALFA(5)*ALFA(5)/TENMPE                                 AX5090
      MI(104)=MI(104)*ALFA(5)                                              AX5100
      MI(105)=MI(105)*ALFA(5)*ALFA(5)                                      AX5110
      SUM=MI(28)+2.0*MI(33)+MI(104)+2.0*MI(105)                            AX5120
      DO 580 I=1,N                                                         AX5130
        SUM=SUM+MI(L5M(I))                                                 AX5140
  580 CONTINUE                                                             AX5150
      MI(5)=CLTOT/(1.0+GAMMA(5)*SUM)                                       AX5160
      ALFA(5)=MI(5)*GAMMA(5)                                               AX5170
      DO 590 I=1,N                                                         AX5180
        MI(L5M(I))=MI(L5M(I))*ALFA(5)                                      AX5190
        ALFA(L5M(I))=MI(L5M(I))*GAMMA(L5M(I))                              AX5200
  590 CONTINUE                                                             AX5210
  600 ALFA(85)=MI(85)*GAMMA(85)                                            AX5220
      ALFA(98)=MI(98)*GAMMA(98)                                            AX5230
      ALFA(27)=AH2O*KW*TENPH                                               AX5240
      MI(27)=ALFA(27)/GAMMA(27)                                            AX5250
      MI(64)=1E0/(TENPH*GAMMA(64))                                         AX5260
      TEST1=S1-ANALCO                                                      AX5270
      TEST2=S2-SO4TOT                                                      AX5280
      TEST3=S3-FTOT                                                        AX5290
      TEST4=S4-PTOT                                                        AX5300
      TEST5=S5-CLTOT                                                       AX5310
      RBIT=0                                                               AX5320
      IF (S1.EQ.0.0.OR.ANALCO.LE.0.0) GO TO 610                            AX5330
      IF (ABS(TEST1).GT.EROR1*ANALCO) RBIT=1                               AX5340
      GO TO 620                                                            AX5350
  610 ANALCO=0.0                                                           AX5360
```

```
  620 IF (S2.EQ.0.0) GO TO 630
      IF (ABS(TEST2).GT.EROR2*SO4TOT) RBIT=1
  630 IF (S3.EQ.0.0) GO TO 640
      IF (ABS(TEST3).GT.EROR3*FTOT) RBIT=1
  640 IF (S4.EQ.0.0) GO TO 650
      IF (ABS(TEST4).GT.EROR4*PTOT) RBIT=1
  650 IF (S5.EQ.0.0) GO TO 660
      IF (ABS(TEST5).GT.EROR5*CLTOT) RBIT=1
  660 IF (PRT(2).NE.0) GO TO 670
      WRITE (6,680) ITER,TEST1,TEST2,TEST3,TEST4,TEST5
  670 RETURN
C
C
  680 FORMAT (' ',3X,I3,5X,5(1PE10.3,2X))
      END
      SUBROUTINE PRINT
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER D,E,DD,RBIT,CORALK,Z(120),LIST4(107),LIST5(8),PRT(4)
      INTEGER PECALC,PECK,ICK(2)
      DOUBLE PRECISION MI(120),KT(200),LOGKT(200),LOGKTO(200),MNTOT,
     *LH2O,MU,NATOT,KTOT,MGTOT,LITOT,NH4TOT,KW,RATIO1(10),RATIO2(10),
     *RATIO3(8),XLGAM(120),AAA(40)
      CHARACTER*1 ASTER,SFG(40)
      CHARACTER*8 NSPEC(120),NREACT(200)
      CHARACTER*80 TITL,PATHLINE,PL2,LINE,FMT
      COMMON MI,KT,LOGKT,LOGKTO,KW,D,E,DD,C,R,T,F,TEMP,A,B,PE,PES,PEDO,
     *PESATO,PECK,PECALC,PH,TENMPE,TENPH,ALFA(120),GAMMA(120),AP(200),
     *XLALFA(120),Z,CUNITS(120),ANALMI(120),GFW(120),DHA(120),NSPEC,
     *NREACT,DH(200),AH2O,LH2O,EROR1,EROR2,EROR3,EROR4,EROR5,EHM,DENS,
     *DOX,XLMI(120),ITER,RBIT,C1SAVE,CORALK,MU,LCHEK(200),CO2TIT,ANALCO,
     *SITOT,CATOT,MGTOT,KTOT,NATOT,SO4TOT,FETOT,PTOT,ALTOT,FTOT,BTOT,
     *LITOT,NH4TOT,SRTOT,BATOT,CLTOT,MNTOT,ICK,PRT,TITL,EPMCAT,EPMAN,
     *NEQU,ISPEC,KSPEC(120),IMIN,KMIN(200),TDS,IDAVES,IPRT,JJ,JK,
     *PATHLINE,PL2,IMBAL
      DATA LIST4/1,2,3,4,64,5,6,7,18,86,27,62,98,19,23,22,21,20,29,32,
     *30,31,49,44,43,42,94,46,95,63,96,93,24,25,26,14,67,68,8,9,10,11,
     *12,77,78,79,80,13,100,65,99,15,16,28,33,34,101,102,106,107,111,
     *109,110,103,104,105,108,112,113,115,51,52,53,54,55,56,57,58,59,60,
     *45,47,48,40,73,41,75,74,76,61,50,36,37,85,38,39,92,81,82,83,88,69,
     *97,89,114,90,91/
      DATA LIST5/1,2,3,4,51,8,6,7/
      CEPMAN=0.0
      CEPMCT=0.0
      ELECT=0.0
      DO 20 I=1,D
        ELECT=ELECT+Z(I)*MI(I)
        IF (Z(I).GT.0) GO TO 10
        CEPMAN=CEPMAN-Z(I)*MI(I)
        GO TO 20
   10   CEPMCT=CEPMCT+Z(I)*MI(I)
   20 CONTINUE
      ELECT=ELECT*1000.
      CEPMAN=CEPMAN*1000.*(1.0-C1SAVE*1.0E-06)
      CEPMCT=CEPMCT*1000.*(1.0-C1SAVE*1.0E-06)
      S1=MI(7)+MI(18)+MI(21)+MI(22)+MI(30)+MI(31)+MI(42)+MI(43)+MI(86)+
     *MI(111)+MI(69)+2.0*MI(97)
      PCO2=0.0
      XLPCO2=-99.9
      IF (S1.GT.0.0) GO TO 30
      GO TO 40
```

```
30    PCO2=ALFA(86)/10.0**(108.3865+0.01985076*T-6919.53/T-40.45154*
     *DLOG10(T)+669365.0/T**2.0)
      XLPCO2=DLOG10(PCO2)
40    EHPE=PE*C*R*T/F
      SUMALK=MI(30)+MI(29)+MI(76)+MI(19)+MI(22)+MI(41)+MI(43)+MI(11)+
     *MI(10)+MI(99)+MI(52)+MI(106)+MI(111)+MI(25)+MI(37)+MI(82)+MI(89)+
     *MI(91)+MI(38)+MI(67)+MI(7)+MI(48)+MI(27)+2.0*(MI(74)+MI(31)+MI(21)
     *+MI(73)+MI(42)+MI(50)+MI(61)+MI(80)+MI(100)+MI(13)+MI(77)+MI(53)+
     *MI(26)+MI(68)+MI(18)+MI(47))+3.0*(MI(75)+MI(40)+MI(12)+MI(78)+
     *MI(107)+MI(115)+MI(45))+4.0*(MI(79)+MI(54))-MI(64)-MI(63)-MI(93)-
     *2.0*MI(96)+MI(69)+2*MI(97)
      SUMALK=SUMALK*1000.
      CARBAL=MI(7)+MI(30)+MI(22)+MI(43)+MI(111)+2.0*(MI(31)+MI(21)+
     *MI(42)+MI(18))+MI(69)+2.0*MI(97)
      CARBAL=CARBAL*1000.
      WRITE (6,180) TITL
      WRITE (6,190)
      WRITE (6,190)
      WRITE (6,200) AH2O,EPMCAT,CEPMCT,PH,PCO2,EPMAN,CEPMAN,XLPCO2,
     *ALFA(70),EHM,PE,TEMP,ALFA(71),PES,S1,PEDO,DENS,PESATO,MU,TDS,
     *SUMALK,CARBAL,ELECT
      WRITE (6,210) PE,EHPE
      WRITE (6,220)
      A0=1000.0
      AAA(22)=SO4TOT*A0
      SFG(22)=PL2(13:13)
      AAA(21)=MI(17)*A0
      SFG(21)=PL2(6:6)
      AAA(3)=CATOT*A0
      SFG(3)=PL2(7:7)
      AAA(4)=ALTOT*A0
      SFG(4)=' '
      AAA(5)=MGTOT*A0
      SFG(5)=PL2(9:9)
      AAA(6)=NATOT*A0
      SFG(6)=PL2(10:10)
      AAA(7)=KTOT*A0
      SFG(7)=PL2(11:11)
      AAA(8)=CLTOT*A0
      SFG(8)=PL2(12:12)
      AAA(9)=FTOT*A0
      SFG(9)=PL2(14:14)
      AAA(10)=SITOT*A0
      SFG(10)=PL2(15:15)
      AAA(11)=MI(98)*A0
      SFG(11)=PL2(16:16)
      AAA(12)=BTOT*A0
      SFG(12)=PL2(17:17)
      AAA(13)=BATOT*A0
      SFG(13)=PL2(18:18)
      AAA(14)=LITOT*A0
      SFG(14)=PL2(19:19)
      AAA(15)=SRTOT*A0
      SFG(15)=PL2(20:20)
      AAA(16)=FETOT*A0
      SFG(16)=PL2(21:21)
      AAA(17)=MNTOT*A0
      SFG(17)=PL2(22:22)
      AAA(18)=(MI(85)+NH4TOT)*A0
      SFG(18)='*'
      IF (PL2(23:23).NE.'*') SFG(18)=' '
```

```
      IF (PL2(24:24).NE.'*') SFG(18)=' '
      AAA(19)=PTOT*A0
      SFG(19)=PL2(25:25)
      AAA(20)=TEMP
      SFG(20)=PL2(1:1)
      AAA(23)=MI(85)*A0
      SFG(23)=PL2(23:23)
      AAA(24)=NH4TOT*A0
      SFG(24)=PL2(24:24)
      AAA(25)=DOX/31.9988
      SFG(25)=' '
      AAA(26)=(MI(7)+MI(22)+MI(30)+MI(43)+MI(69)+MI(111))*A0
      SFG(26)=PL2(4:4)
      AAA(27)=PH
      SFG(27)=PL2(2:2)
      AAA(28)=MI(86)*A0
      SFG(28)=PL2(4:4)
      AAA(29)=(MI(18)+MI(21)+MI(31)+MI(42)+MI(97))*A0
      SFG(29)=PL2(4:4)
      AAA(31)=(MI(8)+MI(11)+MI(12)+MI(34)+MI(65)+MI(80)+MI(100))*A0
      SFG(31)=SFG(16)
      AAA(32)=(MI(9)+MI(10)+MI(13)+MI(15)+MI(16)+MI(28)+MI(33)+MI(77)+
     *MI(78)+MI(79)+MI(99))*A0
      SFG(32)=SFG(16)
      AAA(33)=(MI(101)+MI(103)+MI(104)+MI(105)+MI(106)+MI(107)+MI(108)+
     *MI(109)+MI(110)+MI(111)+MI(115))*A0
      SFG(33)=SFG(17)
      AAA(34)=MI(102)*A0
      SFG(34)=SFG(17)
      AAA(35)=MI(113)*A0
      SFG(35)=SFG(17)
      AAA(36)=MI(112)*A0
      SFG(36)=SFG(17)
      AAA(37)=CUNITS(116)/16.0428
      SFG(37)=PL2(35:35)
      AAA(38)=CUNITS(117)/12.011
      SFG(38)=PL2(26:26)
      AAA(39)=CUNITS(118)
      SFG(39)=PL2(45:45)
      AAA(40)=0
      SFG(40)='*'
C
C Compensate for charge imbalance, if selected
C
      READ (PATHLINE,50) I
      PATHLINE(80:80)=' '
   50 FORMAT (79X,I1)
      IF (I.NE.1) GO TO 80
      SUMPOS=AAA(3)*2+AAA(5)*2+AAA(6)+AAA(7)+AAA(4)*3+AAA(13)*2+AAA(14)+
     *AAA(15)*2+AAA(16)*2+AAA(18)+10**(0D0-PH)
      SUMNEG=AAA(8)+AAA(22)*2+AAA(26)+AAA(9)+AAA(19)+AAA(23)
      IF (SUMNEG*SUMPOS.LE.0.0) GO TO 60
      GMEAN=DSQRT(SUMNEG*SUMPOS)
      FACTNEG=GMEAN/SUMNEG
      FACTPOS=GMEAN/SUMPOS
      AAA(3)=AAA(3)*FACTPOS
      AAA(5)=AAA(5)*FACTPOS
      AAA(6)=AAA(6)*FACTPOS
      AAA(7)=AAA(7)*FACTPOS
      AAA(4)=AAA(4)*FACTPOS
      AAA(13)=AAA(13)*FACTPOS
```

```
      AAA(14)=AAA(14)*FACTPOS
      AAA(15)=AAA(15)*FACTPOS
      AAA(16)=AAA(16)*FACTPOS
      AAA(18)=AAA(18)*FACTPOS
      AAA(9)=AAA(9)*FACTNEG
      AAA(19)=AAA(19)*FACTNEG
      AAA(23)=AAA(23)*FACTNEG
      AAA(8)=AAA(8)*FACTNEG
      AAA(22)=AAA(22)*FACTNEG
      AAA(26)=AAA(26)*FACTNEG
      GO TO 80
   60 WRITE (*,70) TITL(1:32)
   70 FORMAT ('Charge imbalance may still exist: ',A)
C
   80 AAA(30)=AAA(26)+AAA(28)+AAA(29)
      SFG(30)=PL2(4:4)
      AAA(1)=AAA(30)+AAA(37)+AAA(38)
      SFG(1)=PL2(4:4)
      AAA(2)=AAA(21)+AAA(22)
      IF (SFG(21).EQ.'*') THEN
        SFG(2)='*'
      ELSE
        SFG(2)=SFG(22)
      END IF
      WRITE (7,90) TITL(1:32)
   90 FORMAT (A)
      DO 110 I=0,4
        LINE=' '
        DO 100 J=1,8
          FMT='(F8.3,A1)'
          WRITE (LINE(J*9-8:J*9),FMT) AAA(I*8+J),SFG(I*8+J)
  100   CONTINUE
        WRITE (7,90) LINE
  110 CONTINUE
      WRITE (7,90) PATHLINE
C      WRITE (7,155) TITL(1:32),(AAA(INDEX),SFG(INDEX),INDEX=1,39)
C     *   ,PATHLINE
C  155 FORMAT (A32,5(F8.4,A1),4(/,8(F8.4,A1)),/,2(F8.4,A1),/,A80)
      DUM=10.**(-38)
      DO 120 I=1,D
        CUNITS(I)=0.0
        IF (MI(I).LT.DUM) GO TO 120
        CUNITS(I)=MI(I)*1000.*GFW(I)*(1.0-1.0E-06*C1SAVE)
        XLMI(I)=DLOG10(MI(I))
        XLALFA(I)=DLOG10(ALFA(I))
        XLGAM(I)=DLOG10(GAMMA(I))
  120 CONTINUE
      DO 150 I=1,107
        IF (MI(LIST4(I)).LT.DUM) GO TO 150
        IF (ISPEC.EQ.0) GO TO 140
        DO 130 J=1,ISPEC
          IF (LIST4(I).EQ.KSPEC(J)) GO TO 140
  130   CONTINUE
        GO TO 150
  140   WRITE (6,230) LIST4(I),NSPEC(LIST4(I)),Z(LIST4(I)),
     *   CUNITS(LIST4(I)),MI(LIST4(I)),ALFA(LIST4(I)),XLALFA(LIST4(I)),
     *   GAMMA(LIST4(I))
  150 CONTINUE
C
C
C     CALCULATION OF MOLAR RATIOS AND LOG ACTIVITY RATIOS.
```

```
      DO 160 I=1,8
        IF (ANALMI(LIST5(I)).LT.1E-30) ANALMI(LIST5(I))=1E-30
        IF (MI(LIST5(I)).LT.1E-30) MI(LIST5(I))=1E-30
        IF (MI(LIST5(I)).LT.1E-30) XLALFA(LIST5(I))=-30.
        RATIO1(I)=ANALMI(5)/ANALMI(LIST5(I))
        RATIO2(I)=MI(5)/MI(LIST5(I))
  160 CONTINUE
      RATIO1(9)=ANALMI(1)/ANALMI(2)
      RATIO1(10)=ANALMI(3)/ANALMI(4)
      RATIO2(9)=MI(1)/MI(2)
      RATIO2(10)=MI(3)/MI(4)
      RATIO3(1)=XLALFA(1)+PH*2.
      RATIO3(2)=XLALFA(2)+PH*2.
      RATIO3(3)=XLALFA(3)+PH
      RATIO3(4)=XLALFA(4)+PH
      RATIO3(5)=XLALFA(51)+PH*3.
      RATIO3(6)=XLALFA(8)+PH*2.
      RATIO3(7)=XLALFA(1)-XLALFA(2)
      RATIO3(8)=XLALFA(3)-XLALFA(4)
      IF (PRT(3).NE.0) GO TO 170
      WRITE (6,190)
      WRITE (6,240) (RATIO1(I),RATIO2(I),RATIO3(I),I=1,8),(RATIO1(I),
     *RATIO2(I),I=9,10)
  170 WRITE (6,180) TITL
      RETURN
C
C
  180 FORMAT (/1X,A80)
  190 FORMAT (//)
  200 FORMAT (//,22X,'****DESCRIPTION OF SOLUTION ****',//,10X,'ANAL.',
     *5X,'COMP.',9X,'PH',11X,'ACTIVITY H2O = ',F7.4,/,1X,'EPMCAT ',F8.2,
     *2X,F8.2,6X,F6.3,9X,'PCO2= ',1PE13.6,/,1X,'EPMAN  ',0PF8.2,2X,F8.2,
     *21X,'LOG PCO2 = ',F8.4,/,30X,'TEMPERATURE',6X,'PO2 = ',1PE13.6,/,
     *1X,'EH = ',0PF6.4,2X,'PE = ',F7.3,4X,F6.2,' DEG C',5X,'PCH4 = ',
     *1PE13.6,/,1X,'PE CALC S = ',0PF8.3,26X,'CO2 TOT = ',1PE13.6,/,1X,'
     *PE CALC DOX=',0PF7.3,9X,'IONIC STRENGTH',4X,'DENSITY = ',F8.4,/,
     *1X,'PE SATO DOX=',F7.3,9X,1PE13.6,5X,'TDS = '0PF9.1,'MG/L',/,1X,'T
     *OT ALK = ',1PE10.3,' MEQ',22X,'CARB ALK',' = ',E10.3,' MEQ',/,1X,'
     *ELECT   = ',E10.3,' MEQ',/)
  210 FORMAT (1X,'IN COMPUTING THE DISTRIBUTION OF SPECIES,'/,1X,'PE = '
     *,F7.3,5X,'EQUIVALENT EH =',F7.3,'VOLTS',//)
  220 FORMAT (///,25X,'-----------------------',/,25X,'DISTRIBUTION OF S
     *PECIES',/,25X,'-----------------------',//,2X,'I',2X,'SPECIES',9X,
     *'PPM',7X,'MOLALITY',3X,'ACTIVITY',4X,'LOG ACT',4X,'GAMMA',/)
  230 FORMAT (' ',I3,1X,A8,I3,1X,1PE11.4,1X,E11.4,1X,1PE11.4,1X,0PF7.3,
     *1X,1PE11.4)
  240 FORMAT (//,3X,'MOLE RATIOS FROM',9X,'MOLE RATIOS FROM',/,2X,'ANALY
     *TICAL MOLALITY',7X,'COMPUTED MOLALITY',7X,'LOG ACTIVITY',' RATIOS'
     *,/,1X,3(21('-'),4X),/,1X,'CL/CA   = ',1PE11.4,4X,'CL/CA','   = ',
     *E11.4,4X,'LOG CA/H2 = ',0PF9.4,/1X,'CL/MG   = ',1PE11.4,4X,'CL/MG
     *  = ',E11.4,4X,'LOG MG/H2 = ',0PF9.4,/1X,'CL/NA   = ',1PE11.4,4X,'
     *CL/NA   = ',E11.4,4X,'LOG NA/H1 = ',0PF9.4,/,1X,'CL/K    = ',
     *1PE11.4,4X,'CL/K    = ',E11.4,4X,'LOG K/H1  = ',0PF9.4,/,1X,'CL/AL
     *   = ',1PE11.4,4X,'CL/AL   = ',E11.4,4X,'LOG AL/H3 = ',0PF9.4,/,
     *1X,'CL/FE   = ',1PE11.4,4X,'CL/FE   = ',E11.4,4X,'LOG FE/H2 = ',
     *0PF9.4,/,1X,'CL/SO4  = ',1PE11.4,4X,'CL/SO4  = ',E11.4,4X,'LOG CA/
     *MG = ',0PF9.4,/,1X,'CL/HCO3 = ',1PE11.4,4X,'CL/HCO3 = ',E11.4,4X,'
     *LOG NA/K  = ',0PF9.4,/,1X,'CA/MG   = ',1PE11.4,4X,'CA/MG   = ',
     *E11.4,/,1X,'NA/K    = ',E11.4,4X,'NA/K    = ',E11.4)
      END
      SUBROUTINE SAT
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER D,E,DD,RBIT,CORALK,Z(120),LIST6(24),PRT(4)
      INTEGER PECALC,PECK,ICK(2)
      DIMENSION LIST7(101), LIST8(15), LIST9(25), LIST0(101)
      DIMENSION XLRATP(200)
      DOUBLE PRECISION MI(120),KT(200),LOGKT(200),LOGKTO(200),MNTOT,
     *LH2O,MU,NATOT,KTOT,MGTOT,LITOT,NH4TOT,KW
      CHARACTER*8 NSPEC(120),NREACT(200)
      CHARACTER*80 TITL,PATHLINE,PL2
      COMMON MI,KT,LOGKT,LOGKTO,KW,D,E,DD,C,R,T,F,TEMP,A,B,PE,PES,PEDO,
     *PESATO,PECK,PECALC,PH,TENMPE,TENPH,ALFA(120),GAMMA(120),AP(200),
     *XLALFA(120),Z,CUNITS(120),ANALMI(120),GFW(120),DHA(120),NSPEC,
     *NREACT,DH(200),AH2O,LH2O,EROR1,EROR2,EROR3,EROR4,EROR5,EHM,DENS,
     *DOX,XLMI(120),ITER,RBIT,C1SAVE,CORALK,MU,LCHEK(200),CO2TIT,ANALCO,
     *SITOT,CATOT,MGTOT,KTOT,NATOT,SO4TOT,FETOT,PTOT,ALTOT,FTOT,BTOT,
     *LITOT,NH4TOT,SRTOT,BATOT,CLTOT,MNTOT,ICK,PRT,TITL,EPMCAT,EPMAN,
     *NEQU,ISPEC,KSPEC(120),IMIN,KMIN(200),TDS,IDAVES,IPRT,JJ,JK,
     *PATHLINE,PL2,IMBAL
      DATA LIST6/1,2,3,4,5,6,7,8,9,11,18,24,27,40,45,47,51,54,62,67,88,
     *90,101,102/
      DATA LIST7/40,41,141,51,43,18,114,42,22,151,145,49,53,20,13,144,
     *98,50,21,30,57,100,29,12,56,113,120,97,63,28,52,111,112,119,19,65,
     *48,109,118,39,96,46,47,44,129,148,68,99,110,11,108,64,116,117,58,
     *67,59,61,150,55,45,142,115,54,102,37,10,101,147,143,38,66,62,32,
     *60,107,146,154,155,156,172,173,174,175,176,177,178,179,180,181,
     *183,184,185,186,187,188,189,190,191,192,193/
      DATA LIST8/107,108,109,110,111,112,113,114,115,119,120,173,174,
     *175,177/
      DATA LIST9/13,12,10,181,19,144,145,96,107,63,113,111,109,52,174,
     *180,100,101,98,102,47,154,37,120,68/
C
C
C     CALCULATION OF ION ACTIVITY PRODUCTS
      DO 20 I=1,24
        IF (ALFA(LIST6(I)).LT.1.E-38) GO TO 10
        ALFA(LIST6(I))=DLOG10(ALFA(LIST6(I)))
        GO TO 20
   10   ALFA(LIST6(I))=-2E4
   20 CONTINUE
      AP(10)=ALFA(8)+ALFA(18)
      AP(11)=ALFA(2)+ALFA(18)
      AP(12)=ALFA(1)+AP(11)+ALFA(18)
      AP(13)=ALFA(1)+ALFA(18)
      AP(18)=ALFA(1)+ALFA(6)
      AP(19)=AP(18)+2E0*LH2O
      AP(20)=ALFA(2)+2E0*ALFA(27)
      AP(21)=3E0*ALFA(2)+2E0*ALFA(24)+6E0*ALFA(27)-5E0*LH2O
      AP(22)=AP(13)
      AP(28)=2E0*ALFA(2)+ALFA(24)+4E0*(ALFA(27)-LH2O)
      AP(29)=ALFA(1)+ALFA(2)+2E0*ALFA(24)+4E0*ALFA(27)-6E0*LH2O
      AP(30)=ALFA(2)+ALFA(24)+2E0*ALFA(27)-3E0*LH2O
      AP(32)=2E0*ALFA(1)+5E0*ALFA(2)+8E0*ALFA(24)+14E0*ALFA(27)-22E0*
     *LH2O
      AP(37)=2E0*ALFA(2)+3E0*ALFA(24)+4E0*ALFA(27)-4.5E0*LH2O
      AP(38)=3E0*ALFA(2)+4E0*ALFA(24)+6E0*ALFA(27)-1E1*LH2O
      AP(39)=5E0*ALFA(2)+4E0*ALFA(18)+2E0*ALFA(27)+4E0*LH2O
      AP(40)=ALFA(4)+ALFA(54)+3E0*ALFA(24)-8E0*LH2O
      AP(41)=AP(40)-ALFA(4)+ALFA(3)
      AP(42)=ALFA(1)+2E0*(ALFA(54)+ALFA(24))-8E0*LH2O
      AP(43)=ALFA(3)+ALFA(54)+2E0*ALFA(24)-5E0*LH2O
      AP(44)=ALFA(4)+3E0*(ALFA(54)+ALFA(24))-2E0*PH-12E0*LH2O
```

```
      AP(45)=ALFA(4)+ALFA(54)+3E0*(ALFA(2)+ALFA(24))+6E0*ALFA(27)-1E1*
     *LH2O
      AP(46)=.6E0*ALFA(4)+.25E0*ALFA(2)+2.3E0*ALFA(54)+3.5E0*ALFA(24)-1.
     *2E0*PH-11.2E0*LH2O
      AP(47)=2E0*(ALFA(54)+ALFA(24)-PH)-7E0*LH2O
      AP(48)=AP(47)
      C1=(SQRT(MI(1)*GAMMA(1)+MI(2)*GAMMA(2)+MI(3)*GAMMA(3)))
      IF (C1.GT.0.0) C1=DLOG10(C1)
      IF (C1.LE.0.0) C1=-2E4
      AP(49)=.33E0*C1+2.33E0*ALFA(54)+3.67E0*ALFA(24)-2E0*PH-12E0*LH2O
      AP(50)=5E0*ALFA(2)+2E0*ALFA(54)+3E0*ALFA(24)+8E0*ALFA(27)-1E1*
     *LH2O
      AP(51)=ALFA(4)+3E0*ALFA(51)+6E0*ALFA(27)+2E0*ALFA(6)
      AP(52)=ALFA(51)+3E0*ALFA(27)
      AP(53)=AP(52)-LH2O
      AP(54)=2E0*ALFA(54)+4E0*ALFA(24)-2E0*PH-12.0*LH2O
      AP(55)=.5E0*(ALFA(3)+ALFA(4))+ALFA(54)+3E0*ALFA(24)-7E0*LH2O
      AP(56)=ALFA(3)+ALFA(54)+3.5E0*ALFA(24)-6E0*LH2O
      C2=(MI(3)*GAMMA(3)+MI(4)*GAMMA(4))
      IF (C2.GT.0.0) C2=DLOG10(C2)
      IF (C2.LE.0.0) C1=-2E4
      AP(57)=.5E0*C2+ALFA(54)+5E0*ALFA(24)-8.5E0*LH2O
      AP(58)=.5E0*C2+ALFA(54)+4.5E0*ALFA(24)-8E0*LH2O
      AP(59)=ALFA(3)+ALFA(7)
      AP(60)=3E0*ALFA(3)+ALFA(7)+ALFA(18)+2E0*LH2O
      AP(62)=2E0*ALFA(3)+ALFA(18)+LH2O
      AP(61)=AP(62)+9E0*LH2O
      AP(63)=ALFA(1)+2E0*ALFA(62)
      AP(64)=.167E0*ALFA(1)+2.33E0*ALFA(54)+3.67E0*ALFA(24)-2E0*PH-12E0*
     *LH2O
      AP(65)=ALFA(3)+ALFA(5)
      AP(66)=2E0*ALFA(3)+ALFA(6)
      AP(67)=AP(66)+1E1*LH2O
      AP(68)=ALFA(8)+ALFA(67)+PH
      AP(96)=5E0*ALFA(1)+3E0*(ALFA(47)-LH2O)+4E0*ALFA(27)
      AP(97)=5E0*ALFA(1)+3E0*(ALFA(47)-LH2O)+3E0*ALFA(27)+ALFA(62)
      AP(98)=ALFA(24)-2E0*LH2O
      AP(99)=ALFA(4)+7E0*ALFA(24)+PH-9E0*LH2O
      AP(100)=AP(98)
      AP(101)=AP(98)
      AP(102)=AP(98)
      IF (ABS(PE).LT.20.0) GO TO 30
      GO TO 40
   30 AP(107)=3E0*ALFA(8)+2E0*ALFA(45)+8E0*LH2O
      AP(108)=3E0*ALFA(9)-PE+4E0*LH2O+8E0*PH
      AP(109)=2E0*ALFA(9)+3E0*LH2O+6E0*PH
      AP(110)=AP(109)
      AP(111)=ALFA(9)+3E0*ALFA(27)-LH2O
      AP(112)=3E0*ALFA(8)+2E0*ALFA(24)+6E0*ALFA(27)-5E0*LH2O
      AP(113)=ALFA(9)+3E0*(LH2O+PH)
      AP(114)=AP(45)+3E0*(ALFA(8)-ALFA(2))
      AP(115)=ALFA(8)+2E0*(ALFA(67)+PE+PH)
      AP(119)=3E0*ALFA(8)+4E0*ALFA(67)+2E0*PE+4E0*PH
      AP(120)=AP(68)
      AP(173)=ALFA(102)+2*LH2O+4*PH+PE
      AP(174)=AP(173)
      AP(175)=AP(173)
      AP(177)=3*ALFA(101)+4*LH2O+8*PH+2*PE
      GO TO 60
   40 DO 50 I=1,15
         J1=LIST8(I)
```

```
        AP(J1)=-6000.                                                   AZ1240
   50 CONTINUE                                                          AZ1250
      PECK=1                                                            AZ1260
   60 AP(116)=.29*ALFA(2)+.23*ALFA(9)+1.58*ALFA(54)+3.93*ALFA(24)-10.*  AZ1270
     *LH2O                                                              AZ1280
      AP(117)=.45*ALFA(2)+.34*ALFA(9)+1.47*ALFA(54)+3.82*ALFA(24)-9.2*  AZ1290
     *LH2O+.76*PH                                                       AZ1300
      AP(118)=3E0*ALFA(2)+ALFA(1)+4E0*ALFA(18)                          AZ1310
      AP(129)=ALFA(1)+2E0*ALFA(54)+4E0*ALFA(24)-8E0*LH2O                AZ1320
      AP(141)=AP(52)                                                    AZ1330
      AP(142)=2E0*(ALFA(1)+ALFA(54)+PH)+3E0*ALFA(24)-8E0*LH2O           AZ1340
      AP(143)=ALFA(88)+ALFA(18)                                         AZ1350
      AP(144)=ALFA(88)+ALFA(6)                                          AZ1360
      AP(145)=ALFA(90)+ALFA(6)                                          AZ1370
      AP(146)=ALFA(90)+ALFA(18)                                         AZ1380
      AP(147)=ALFA(9)+ALFA(45)+2E0*LH2O                                 AZ1390
      AP(148)=2E0*ALFA(1)+4E0*ALFA(54)+8E0*ALFA(24)-17E0*LH2O           AZ1400
      AP(150)=ALFA(2)+ALFA(18)+3E0*LH2O                                 AZ1410
      AP(151)=2E0*ALFA(1)+ALFA(18)+2E0*ALFA(27)+3E0*LH2O                AZ1420
      AP(172)=ALFA(101)+LH2O+2*PH                                       AZ1430
      AP(176)=2*ALFA(102)+3*LH2O+6*PH                                   AZ1440
      AP(178)=ALFA(101)+2*ALFA(27)                                      AZ1450
      AP(179)=ALFA(102)+3*ALFA(27)                                      AZ1460
      AP(180)=ALFA(102)+2*LH2O+3*PH                                     AZ1470
      AP(181)=ALFA(101)+ALFA(18)                                        AZ1480
      AP(183)=ALFA(101)+2*ALFA(5)                                       AZ1490
      AP(184)=AP(183)+LH2O                                              AZ1500
      AP(185)=AP(183)+2*LH2O                                            AZ1510
      AP(186)=AP(183)+4*LH2O                                            AZ1520
      AP(187)=2*ALFA(101)+ALFA(24)+4*PH                                 AZ1530
      AP(188)=2*ALFA(101)+ALFA(24)+2*PH-LH2O                            AZ1540
      AP(189)=ALFA(101)+ALFA(67)+PH                                     AZ1550
      AP(190)=ALFA(101)+ALFA(6)                                         AZ1560
      AP(191)=2*ALFA(102)+3*ALFA(6)                                     AZ1570
      AP(192)=3*ALFA(101)+2*ALFA(45)                                    AZ1580
      AP(193)=ALFA(101)+ALFA(47)                                        AZ1590
      AP(154)=AP(37)                                                    AZ1600
      AP(155)=AP(52)-LH2O                                               AZ1610
      AP(156)=AP(129)-2*LH2O                                            AZ1620
      WRITE (6,190)                                                     AZ1630
      WRITE (6,200)                                                     AZ1640
      DO 70 I=1,200                                                     AZ1650
   70   XLRATP(I)=99.999                                                AZ1660
      IF (PRT(4).NE.9) GO TO 90                                         AZ1670
      INPRINT=25                                                        AZ1680
      DO 80 I=1,INPRINT                                                 AZ1690
        LIST0(I)=LIST9(I)                                               AZ1700
   80 CONTINUE                                                          AZ1710
      GO TO 110                                                         AZ1720
   90 INPRINT=101                                                       AZ1730
      DO 100 I=1,INPRINT                                                AZ1740
        LIST0(I)=LIST7(I)                                               AZ1750
  100 CONTINUE                                                          AZ1760
  110 DO 150 I=1,INPRINT                                                AZ1770
        IF (IMIN.EQ.0) GO TO 130                                        AZ1780
        K=0                                                             AZ1790
        DO 120 J=1,IMIN                                                 AZ1800
          IF (LIST0(I).EQ.KMIN(J)) K=1                                  AZ1810
  120   CONTINUE                                                        AZ1820
        IF (K.EQ.1) GO TO 130                                           AZ1830
        GO TO 150                                                       AZ1840
```

```
  130   IF (AP(LISTO(I)).LT.-38.0.OR.AP(LISTO(I)).GT.38.0) GO TO 140    AZ1850
        IF (LCHEK(LISTO(I)).EQ.1) GO TO 140                             AZ1860
        DUM=AP(LISTO(I))-DLOG10(KT(LISTO(I)))                           AZ1870
        IF (DUM.GT.75.) GO TO 140                                       AZ1880
        XIAP=10.**AP(LISTO(I))                                          AZ1890
        RAT=XIAP/KT(LISTO(I))                                           AZ1900
        XLRAT=DLOG10(RAT)                                               AZ1910
        DELGR=C*R*T*XLRAT                                               AZ1920
        WRITE (6,210) LISTO(I),NREACT(LISTO(I)),XIAP,KT(LISTO(I)),      AZ1930
     *   AP(LISTO(I)),LOGKT(LISTO(I)),RAT,XLRAT                         AZ1940
        GO TO 150                                                       AZ1950
  140   IF (AP(LISTO(I)).LT.-5000.0.OR.AP(LISTO(I)).GT.5000.0) GO TO    AZ1960
     *   150                                                            AZ1970
        XLRAT=AP(LISTO(I))-LOGKT(LISTO(I))                              AZ1980
        DELGR=C*R*T*XLRAT                                               AZ1990
        WRITE (6,220) LISTO(I),NREACT(LISTO(I)),AP(LISTO(I)),           AZ2000
     *   LOGKT(LISTO(I)),XLRAT                                          AZ2010
  150 CONTINUE                                                          AZ2020
      IF (PECK.EQ.1.AND.PECALC.NE.0) GO TO 160                          AZ2030
      GO TO 180                                                         AZ2040
  160 WRITE (6,230)                                                     AZ2050
      DO 170 I=1,15                                                     AZ2060
        WRITE (6,240) NREACT(LIST8(I))                                  AZ2070
  170 CONTINUE                                                          AZ2080
  180 RETURN                                                            AZ2090
C                                                                       AZ2100
C                                                                       AZ2110
  190 FORMAT (//)                                                       AZ2120
  200 FORMAT (//,6X,'PHASE',6X,'IAP',7X,'KT',6X,'LOG IAP',2X,'LOG KT',  AZ2130
     *4X,'IAP/KT',2X,'LOG IAP/KT',/)                                    AZ2140
  210 FORMAT (' ',I3,1X,A8,2(1PE10.3),2(1X,0PF8.3),1X,1PE10.3,1X,0PF8.3) AZ2150
  220 FORMAT (' ',I3,1X,A8,21X,2(F8.3,1X),9X,F10.3)                     AZ2160
  230 FORMAT (///,1X,'PE IS GREATER THAN 20 OR LESS THAN -20',/,1X,'AN', AZ2170
     *'D THE FOLLOWING MINERAL REACTIONS HAVE BEEN DISREGARDED',/)      AZ2180
  240 FORMAT (' ',20X,A8)                                               AZ2190
      END                                                               AZ2200
```

ATTACHMENT B:

Listing of data file to WATEQFP; WATEQF.DAT

```
'CA       '  2    40.0800  6.0
'MG       '  2    24.3120  6.5
'NA       '  1    22.9898  4.0
'K        '  1    39.1020  3.0
'CL       ' -1    35.4530  3.0
'SO4      ' -2    96.0616  4.0
'HCO3     ' -1    61.0173  5.4
'FE       '  2    55.8470  6.0
'FE       '  3    55.8470  9.0
'FEOH     '  2    72.8544  5.0
'FEOH     '  1    72.8549  5.0
'FE(OH)3  ' -1   106.8690  5.0
'FEHPO4   '  1   151.8200  5.4
'H2S AQ   '  0    34.0799  0.0
'FESO4    '  1   151.9086  5.0
'FECL     '  2    91.3000  5.0
'ANAL H2S '  0    34.0799  0.0
'CO3      ' -2    60.0094  5.4
'MGOH     '  1    41.3194  6.5
'MGF      '  1    43.3104  4.5
'MGCO3 AQ '  0    84.3214  0.0
'MGHCO3   '  1    85.3293  4.0
'MGSO4 AQ '  0   120.3736  0.0
'H4SIO4AQ '  0    96.1155  0.0
'H3SIO4   ' -1    95.1075  4.0
'H2SIO4   ' -2    94.0995  5.4
'OH       ' -1    17.0074  3.5
'FECL2    '  1   126.7530  5.0
'CAOH     '  1    57.0874  6.0
'CAHCO3   '  1   101.0973  6.0
'CACO3 AQ '  0   100.0890  0.0
'CASO4 AQ '  0   136.1416  0.0
'FECL3    '  0   162.2060  0.0
'FESO4    '  0   151.9086  0.0
'SIO2 TOT '  0    60.0848  0.0
'H3BO3 AQ '  0    61.8331  0.0
'H2BO3    ' -1    60.8251  2.5
'NH3   AQ '  0    17.0306  0.0
'NH4-N    '  1    14.0067  2.5
'MGPO4    ' -1   119.2834  5.4
'MGH2PO4  '  1   121.2993  5.4
'NACO3    ' -1    82.9992  5.4
'NAHCO3   '  0    83.9909  0.0
'NASO4    ' -1   119.0514  5.4
'PO4-P    ' -3    30.9738  5.0
'KSO4     ' -1   135.1636  5.4
'HPO4     ' -2    95.9794  5.0
'H2PO4    ' -1    96.9873  5.4
'CAF+     '  1    59.0784  5.0
'NAHPO4   ' -1   118.9692  5.4
'AL       '  3    26.9815  9.0
'ALOH     '  2    43.9889  5.4
'AL(OH)2  '  1    60.9962  5.4
```

```
'AL(OH)4  '  -1      95.0110  4.5
'ALF      '   2      45.9799  5.4
'ALF2     '   1      64.9783  5.4
'ALF3     '   0      83.9767  0.0
'ALF4     '  -1     102.9751  4.5
'ALSO4    '   1     123.0431  4.5
'AL(SO4)2 '  -1     219.1047  4.5
'KHPO4    '  -1     135.0814  5.4
'F        '  -1      18.9984  3.5
'HSO4     '  -1      97.0696  4.5
'H        '   1       1.0080  9.0
'FEH2PO4  '   1     152.8340  5.4
'H2S CALC '   0      34.0799  0.0
'HS       '  -1      33.0720  3.5
'S        '  -2      32.0640  5.0
'SRHCO3   '   1     148.6373  5.4
'PO2      '   0      31.9988  0.0
'PCH4     '   0      16.0430  0.0
'AH2O     '   0      18.0153  0.0
'MGHPO4   '   0     120.2914  0.0
'CAHPO4   '   0     136.0594  0.0
'CAPO4    '  -1     135.0514  5.4
'CAH2PO4  '   1     137.0673  5.4
'FE(OH)2  '   1      89.8616  5.4
'FE(OH)3  '   0     106.8689  0.0
'FE(OH)4  '  -1     123.8762  5.4
'FE(OH)2  '   0      89.8616  0.0
'LI       '   1       6.9390  6.0
'LIOH     '   0      23.9464  0.0
'LISO4    '  -1     103.0006  5.0
'NH4CALC  '   1      18.0386  2.5
'NO3-N    '  -1      14.0067  3.0
'H2CO3    '   0      62.0253  0.0
'B TOT    '   0      10.8100  0.0
'SR       '   2      87.6200  5.0
'SROH     '   1     104.6274  5.0
'BA       '   2     137.3400  5.0
'BAOH     '   1     154.3474  5.0
'NH4SO4   '  -1     114.1002  5.0
'HCL      '   0      36.4610  0.0
'NACL     '   0      58.4428  0.0
'KCL      '   0      74.5550  0.0
'H2SO4    '   0      98.0775  0.0
'SRCO3    '   0     147.6294  0.0
'BR       '  -1      79.9090  4.0
'FEH2PO4  '   2     152.8340  5.4
'FEHPO4   '   0     151.8200  0.0
'MN       '   2      54.9400  6.0
'MN       '   3      54.9400  9.0
'MNCL     '   1      90.3970  5.0
'MNCL2    '   0     125.8540  0.0
'MNCL3    '  -1     161.3110  5.0
'MNOH     '   1      71.8480  5.0
'MN(OH)3  '  -1     105.9640  5.0
'MNF      '   1      73.9400  5.0
'MNSO4    '   0     151.0060  0.0
'MN(NO3)2 '   0     178.9560  0.0
'MNHCO3   '   1     115.9590  5.0
'MNO4     '  -1     118.9400  3.0
'MNO4     '  -2     118.9400  5.0
'SRSO4    '   0     183.6800  0.0
```

```
'HMNO2    '  -1      87.9480  5.0
'KFE +3   '      9.7000  -13.038
'KFEH+2   '     20.1150  -15.228
'KFEOH+   '     13.2180   -9.500
'KFEOH3   '     30.3     -31.000
'KFESO4   '     13.61     -9.118
'KFECL    '     18.1520  -11.6000
'KFECL2   '      0.0     -10.9190
'KFECL3   '      0.0     -11.9250
'KFESO    '      3.23      2.25
'SIDERITE '     -6.14    -10.57
'MAGNESIT '     -6.1690   -8.2400
'DOLOMITE '     -9.4360  -17.09
'CALCITE  '     -2.2970   -8.4800
'KH3SIO4  '      8.9350   -9.9290
'KH2SIO4  '     29.7170  -21.6170
'KHPO4    '     -3.5300   12.3460
'KH2PO4   '     -4.5200   19.5530
'ANHYDRIT '     -4.3      -4.384
'GYPSUM   '     -0.028    -4.602
'BRUCITE  '      0.8500  -11.4100
'CHRYSOTL '     27.5850  -51.8000
'ARAGONIT '     -2.5890   -8.3360
'KMGF     '      4.6740    1.8200
'KCASO4   '      1.5000    2.3090
'KMGOH    '      2.0900    2.2100
'KH3BO3   '      3.2190   -9.2350
'KNH3     '     12.4770   -9.2440
'FORSTRIT '      4.8700  -28.1100
'DIOPSIDE '     21.1000  -36.2200
'CLENSTIT '      6.6750  -16.8700
'KNAHPO   '      0.0       0.2900
'TREMOLIT '     90.2150 -140.3000
'KKHPO4   '      0.0       0.2900
'KMGHPO4  '      3.3000    2.8700
'KCAHPO4  '      3.3000    2.7390
'KH2CO3   '     -2.1770    6.3520
'SEPIOLIT '     26.5320  -40.1000
'TALC     '     45.0650  -62.2900
'HYDMAG   '    -25.5200  -37.8200
'ADULAR   '     30.8200  -20.5700
'ALBITE   '     25.8960  -18.0000
'ANORTH   '     17.5300  -19.3300
'ANALCM   '     18.2060  -12.7000
'KMICA    '     67.8600  -49.0900
'PHLOG    '      0.0     -63.5300
'ILLITE   '     54.6840  -40.3100
'KAOLIN   '     49.1500  -36.9100
'HALLOY   '     44.6800  -32.8200
'BEIDEL   '     60.3550  -45.2600
'CHLOR    '     54.7600  -90.6100
'ALUNIT   '     29.8200  -85.3200
'GIBCRS   '     14.4700  -32.7700
'BOEHM    '     11.9050  -33.4100
'PYROPH   '      0.0     -42.4300
'PHILIP   '      0.0     -19.8600
'ERION    '      0.0       0.0
'CLINOP   '      0.0       0.0
'MORDEN   '      0.0       0.0
'NAHCOL   '      3.7200   -0.5480
'TRONA    '    -18.0000   -0.7950
```

```
'NATRON  '      15.7450   -1.3110
'THRNAT  '      -2.8020    0.1250
'FLUOR   '       4.71    -10.96
'MONTCA  '      58.3730  -45.0000
'HALITE  '       0.9180    1.5820
'THENAR  '      -0.5720   -0.1790
'MIRABI  '      18.9870   -1.1130
'MACKIT  '       0.0      -4.6310
'KHCO3   '      -3.5610   10.329
'KNACO3  '       8.9110    1.2680
'KNAHCO3 '       0.0      -0.2500
'KNASO4  '       1.12      0.7200
'KKSO4   '       2.25      0.8470
'KMGCO3  '       2.7100    2.9800
'KMGHCO3 '       1.0770    1.0660
'KMGSO4  '       4.6       2.2380
'KCAOH   '       1.1900    1.4000
'KCAHCO3 '       4.1100    1.0950
'KCACO3  '       3.5560    3.2240
'KCAF+   '       4.1200    0.9400
'KALOH   '       1.43      9.030
'KALOH2  '       0.0      18.700
'KALOH4  '     -11.16     33.00
'KALF    '       0.0       7.0100
'KALF2   '      20.0000   12.7500
'KALF3   '       2.5000   17.0200
'KALF4   '       0.0      19.7200
'KALSO4  '       2.15      3.02
'KASO42  '       2.84      4.92
'KHSO4   '       4.91      1.987
'KH2SC   '     -65.4400   40.6440
'KH2S    '       5.2990   -6.9420
'KHS     '      12.1000  -12.9180
'KOXY    '      34.1570  -20.7800
'KCH4    '     -57.4350   30.7410
'HYXAPT  '      17.2250  -59.3500
'FLUAPT  '      19.6950  -66.7900
'CHALC   '       4.6150   -3.5230
'MAGADI  '       0.0     -14.3000
'SILGEL  '       5.5000   -2.70
'SILGLAS '       4.4400   -3.0170
'QUARTZ  '       6.2200   -4.0050
'KFEOH2  '       0.0     -18.708
'KFEOH3  '       0.0     -26.638
'KFEOH4  '       0.0     -34.638
'KFEOH2  '      28.5650  -20.5700
'VIVIAN  '       0.0     -36.0000
'MAGNET  '     -40.6600   -9.5650
'HEMATI  '     -30.8450   -4.0070
'MAGHEM  '       0.0       6.3700
'GOETH   '      25.5550  -44.197
'GREENA  '       0.0     -63.1900
'FEOH3A  '       0.0       4.8850
'ANNITE  '      62.4800  -84.2400
'PYRITE  '      11.3000  -18.4800
'MONTBF  '       0.0     -34.9700
'MONTAB  '       0.0     -29.7800
'HUNTITE '     -25.7600  -30.5100
'GREGITE '       0.0     -17.9700
'FESPPT  '       0.0      -3.9150
'KFEH2P  '       0.0       2.7000
```

```
'KCAPO4   '      3.1000    6.4590
'KCAH2P   '      3.4000    1.4080
'KMGPO4   '      3.1000    6.5890
'KMGH2P   '      3.4000    1.5130
'KLIOH    '      4.8320    0.2000
'KLISO4   '      0.0       0.6400
'KNH4R    '   -187.0550  119.0770
'LAUMON   '     39.6100  -30.9600
'KSROH    '      1.1500    0.8200
'KBAOH    '      1.7500    0.6400
'KNH4SO   '      0.0       1.1100
'KHCL     '      0.0     -30.0
'KNACL    '      0.0     -30.0
'KKCL     '      0.0     -30.0
'KH2SO4   '      0.0     -30.0
'KO2 SATO '      0.0     -11.3850
'KCO2     '     -4.776    -1.468
'KFEHPO   '      0.0       3.6000
'KFEHP+   '      0.0      -7.6130
'ALOH3A   '     12.9900  -31.6100
'PREHNT   '     10.3900  -11.5200
'STRONT   '     -0.400    -9.271
'CELEST   '      0.228    -6.578
'BARITE   '      6.1410   -9.978
'WITHERIT '      6.9500   -8.585
'STRENGIT '     -2.0300  -26.4000
'LEON     '     90.0700  -69.5700
'KSRHCO3  '      6.05      1.18
'NESQUE   '     -5.789    -5.2110
'ARTIN    '     -1.842   -18.4000
'K O2AQ   '     33.4570  -21.4950
'KW       '     13.3410  -13.9920
'SEP PT   '      0.0     -37.2120
'DIASP    '    -15.4050  -35.0600
'WAIRKT   '     26.1400  -26.6200
'KFEHP2   '      0.0      -7.5830
'KMN 3+   '     25.7600  -25.5070
'KMNCL+   '      0.0       0.6070
'KMNCL2   '      0.0       0.0410
'KMNCL3-  '      0.0      -0.3050
'KMNOH+   '      0.0       3.4490
'KMN(OH)3 '      0.0       7.7820
'KMNF+    '      0.0       0.8500
'KMNSO4   '      3.7000    1.7080
'KMNNO3,2 '     -0.3960    0.0590
'KMNHCO3+ '      0.0       1.7160
'KMNO4-   '    176.6200 -127.8240
'KMNO4--  '    150.0200 -118.4400
'KSRCO3   '      5.22      2.81
'KHMNO2-- '      0.0     -34.4400
'MANGANO  '    -24.0250   17.9380
'PYROLUST '    -29.1800   15.8610
'BIRNSITE '      0.0      18.0910
'NUSTITE  '      0.0      17.5040
'BIXBYITE '    -15.2450   -0.6110
'HAUSMITE '    -80.1400   61.5400
'MNOH2    '      4.1000  -12.9120
'MNOH3    '     20.0900  -35.6440
'MANGANIT '      0.0      -0.2380
'RHODOCHR '     -2.0790  -10.5390
'KSRSO4   '      1.6       2.55
```

```
'MNCL2     '    -17.6220     8.7600
'MNCL2,1W  '    -7.1750      5.5220
'MNCL2,2W  '     1.7100      3.9740
'MNCL2,4W  '    17.3800      2.7100
'TEPHRITE  '    -40.0600    23.1220
'RHODONIT  '    -21.8850     9.5220
'MNS GRN   '    -5.7900      3.8000
'MNSO4     '    -15.4800     2.6690
'MN2SO4,3  '    -39.0600    -5.7110
'MN3PO4,2  '     2.1200    -23.8270
'MNHPO4    '     0.0       -12.9470
```

ATTACHMENT C:

Listing of Source Code to NETPATH; NETPATH.FOR, EP.FOR, PY.FOR

```
      PROGRAM NETPATH
C --------------------------------------
C                NETPATH
C                  by
C           Eric C. Prestemon
C
C    with technical assistance from
C L. Niel Plummer and David L. Parkhurst
C --------------------------------------
C
C This is the main loop of the program. Options are displayed and
C branches are made to the proper subroutines.  Exiting the program
C is also handled, including writing NETPATH.DAT.
C
$INCLUDE:'NETPATH.BLO'
      CALL INITVALS (0)
      CALL WELLFILE
      CALL MODELS
      CALL SCREEN
   10 CALL POSCUR (-1)
      WRITE (*,20)
   20 FORMAT (' Select: <A>dd, <D>elete, <E>dit, <R>un, <S>ave, or ','<Q
     *>uit')
      READ (*,30) ANS
      CALL POSCUR (-1)
      ANS=UPCS(ANS)
      IF (ANS.EQ.'A'.OR.ANS.EQ.'D'.OR.ANS.EQ.'E'.OR.ANS.EQ.'R'.OR.ANS.
     *EQ.'Q'.OR.ANS.EQ.'S') GO TO 80
      GO TO 10
   30 FORMAT (A,A)
   40 CALL POSCUR (-1)
      WRITE (*,50)
   50 FORMAT (' Do you really want to quit? <Enter> for yes.')
      READ (*,30) ANS
      IF (UPCS(ANS).NE.'N') THEN
        OPEN (7,FILE='netpath.dat',STATUS='unknown')
        CLOSE (7,STATUS='DELETE')
        OPEN (7,FILE='netpath.dat',STATUS='new')
        DO 60 J=1,FLIN
   60     WRITE (7,'(a)') FLINE(J)
        CLOSE (7)
        CALL CLS
        WRITE (*,70)
        STOP
      END IF
      GO TO 10
   70 FORMAT (' NETPATH finished - Thank You'/)
   80 IF (ANS.EQ.'D') CALL DELETE
      IF (ANS.EQ.'A') CALL ADD
      IF (ANS.EQ.'Q') GO TO 40
      IF (ANS.EQ.'R') CALL RUN (1)
      IF (ANS.EQ.'E') CALL EDIT
      IF (ANS.EQ.'S') CALL SAVE
```

```
      GO TO 10                                                          AA 540
      END                                                               AA 550
      BLOCK DATA                                                        AB  10
C                                                                       AB  20
C The initial values of arrays contained in the common blocks are set.  AB  30
C                                                                       AB  40
$INCLUDE:'NETPATH.BLO'                                                  AB  50
      DATA MODEL /'Original Data       ',                               AB  60
     *'Mass Balance        ',                                           AB  70
     *'Vogel               ',                                           AB  80
     *'Tamers              ',                                           AB  90
     *'Ingerson and Pearson',                                           AB 100
     *'Mook                ',                                           AB 110
     *'Fontes and Garnier  ',                                           AB 120
     *'Eichinger           ',                                           AB 130
     *'User-defined        '/                                           AB 140
      DATA YES /'No ','Yes'/                                            AB 150
      DATA ION /'Computed  ','50/50     ','Ca/Na     ','Var. Ca/Mg'/    AB 160
      DATA ELESHORT/'C ','S ','CA','AL','MG','NA','K ','CL','F ',       AB 170
     *'SI','BR','B ','BA','LI','SR','FE','MN','N ','P ','RS','I1'       AB 180
     *,'I2','I3','I4','TE','D ','18','TR','I5','I6','I7','I8','EX'/     AB 190
      DATA ELELONG /  '            '                                    AB 200
     *,'Carbon      ','Sulfur      ','Calcium     ','Aluminum    '      AB 210
     *,'Magnesium   ','Sodium      ','Potassium   ','Chloride    '      AB 220
     *,'Fluoride    ','Silica      ','Bromide     ','Boron       '      AB 230
     *,'Barium      ','Lithium     ','Strontium   ','Iron        '      AB 240
     *,'Manganese   ','Nitrogen    ','Phosphorus  ','Redox       '      AB 250
     *,'Carbon-13   ','C-14 (% mod)','Sulfur-34   ','Strontium-87'      AB 260
     *,'Temperature ','Deuterium   ','Oxygen-18   ','Tritium     '/     AB 270
      DATA C14DAT /0.,100.,0.,0.,-25.,100.,100.,0.,0.,0.,0.,0.,0./      AB 280
      DATA FFACT /'Mook  ','Deines, et al.'/                            AB 290
      END                                                               AB 300
      SUBROUTINE ADD                                                    AC  10
C                                                                       AC  20
C The specific add routines are called                                  AC  30
C                                                                       AC  40
$INCLUDE:'NETPATH.BLO'                                                  AC  50
   10 CALL POSCUR (-1)                                                  AC  60
      WRITE (*,20)                                                      AC  70
   20 FORMAT (' Add (1) constraints, (2) phases, or (3) neither?')      AC  80
      READ (*,*,ERR=10) I                                               AC  90
      IF (I.EQ.3) RETURN                                                AC 100
      CALL POSCUR (-1)                                                  AC 110
      IF (I.EQ.1.OR.I.EQ.2) GO TO 30                                    AC 120
      GO TO 10                                                          AC 130
   30 IF (I.EQ.1) CALL ADDCON                                           AC 140
      IEDIT=0                                                           AC 150
      IF (I.EQ.2) CALL ADDPHA (NOPHA+1)                                 AC 160
      CALL SCREEN                                                       AC 170
      RETURN                                                            AC 180
      END                                                               AC 190
      SUBROUTINE ADDCON                                                 AD  10
C                                                                       AD  20
C Additional constraints can be added to the list of constraints to be  AD  30
C used in the model                                                     AD  40
C                                                                       AD  50
$INCLUDE:'NETPATH.BLO'                                                  AD  60
   10 CALL CLS                                                          AD  70
      CALL CONLIST (28)                                                 AD  80
   20 WRITE (*,30)                                                      AD  90
   30 FORMAT (//,' Enter constraint to add (<Enter> to quit,',' ''?'' fo AD 100
```

```
     *r list):')
      READ (*,'(A)',ERR=20) BANS
      IF (BANS.EQ.' ') RETURN
      IF (BANS.EQ.'?') GO TO 10
      READ (BANS,'(I3)',ERR=20) I
      IF (I.GT.28.OR.I.LT.1) GO TO 20
      J=0
   40 J=J+1
      IF (J.GT.NOELE) GO TO 60
      IF (I.NE.IELE(J)) GO TO 40
      WRITE (*,50)
   50 FORMAT (' Constraint already entered.  <Enter> to continue.')
      READ (*,'(a)',ERR=20) ANS
      GO TO 20
   60 NOELE=NOELE+1
      IELE(NOELE)=I
      GO TO 20
      END
      SUBROUTINE ADDPHA (II)
C
C This subroutine is called in two situations.  First, new phases are
C added to the list of phases to be considered in the model.  Second, it
C is called when a phase is to be replaced.  If the phase is edited by
C piece (changing of some constraints of the phase, rather than
C substituting a new phase), EDITPIEC is called.
C
$INCLUDE:'NETPATH.BLO'
      CHARACTER*2 PELEM(15)
      CALL CLS
   10 FORMAT (A80)
   20 IF (FLIN.EQ.1) THEN
        IIOLD=II
        CALL EDITPIEC (II)
        IF (IIOLD.NE.II.AND.IEDIT.EQ.0) GO TO 20
        RETURN
      END IF
   30 WRITE (*,40)
   40 FORMAT (//,' Input phase number (type ''L'' to see phases and',' t
     *heir corresponding numbers.)')
      IF (IEDIT.EQ.0) WRITE (*,50) FLIN
      IF (IEDIT.EQ.1) WRITE (*,60) FLIN,PHASE(II)
   50 FORMAT (I4,' to create new phase, <Enter> to stop entering phases'
     *)
   60 FORMAT (I4,' to edit phase composition, <Enter> for ''',A,'''')
      READ (*,10) LINE
      IF (LINE.EQ.' '.AND.IEDIT.EQ.1) GO TO 170
      IF (LINE.EQ.' ') RETURN
      IF (UPCS(LINE(1:1)).EQ.'L') THEN
        CALL PHALIST (IJ)
        GO TO 70
      END IF
      READ (LINE,'(i3)',ERR=20) IJ
      IF (IEDIT.EQ.1.AND.IJ.EQ.0) RETURN
   70 IF (IJ.LE.0.OR.IJ.GT.FLIN) GO TO 30
      IF (IEDIT.EQ.0) NOPHA=NOPHA+1
      IF (IEDIT.EQ.0) II=NOPHA
      IF (IJ.EQ.FLIN) THEN
        CALL EDITPIEC (II)
        IF (IEDIT.EQ.0) GO TO 20
        RETURN
      END IF
```

```
      F(II)=' '
      READ (FLINE(IJ),80) PHASE(II),IT(II),(PELEM(J),PCOEFF(II,J),J=1,
     *15)
      CALL DONTHAVE (II,1)
      IEDIT=0
   80 FORMAT (A8,1X,A1,15(A2,F8.4))
      DO 90 I=1,15
   90   JELE(II,I)=0
      J=0
      JP=0
  100 JP=JP+1
  110 J=J+1
      IF (PELEM(J).EQ.'  ') JELE(II,JP)=0
      IF (PELEM(J).EQ.'  ') GO TO 170
      K=0
  120 K=K+1
      IF (PELEM(J).EQ.ELESHORT(K)) GO TO 130
      IF (K.LT.33) GO TO 120
      WRITE (*,'(//,a,1x,i2)') ' Bad constraint in phase #',II
      STOP
  130 IF (K.GT.20) GO TO 150
  140 JELE(II,JP)=K
      IF (J.LT.14) GO TO 100
      GO TO 170
  150 IF (K.GT.24.AND.K.LT.29) GO TO 140
      IF (K.LE.24) THEN
        PARA(II,K-19)=PCOEFF(II,JP)
        CALL HAVE (II,K-19)
      END IF
      IF (K.GE.29.AND.K.LE.32) THEN
        PARA(II,K-23)=PCOEFF(II,JP)
        CALL HAVE (II,K-23)
      END IF
      DO 160 K=JP,14
  160   PCOEFF(II,K)=PCOEFF(II,K+1)
      GO TO 110
  170 CALL TRANS (II)
      IF (IEDIT.EQ.0) GO TO 20
      RETURN
      END
      SUBROUTINE BALN
C
C  The mass balance solutions for the models are computed. Various
C  subroutines to set up the combinations of constraints and phases
C  and print the solutions are called.
C
$INCLUDE:'NETPATH.BLO'
      CHARACTER*8 TEST
      CALL CLS
      WUNIT=8
      TUNIT=6
      IUNIT=TUNIT
      IF (NRUN.EQ.0) IUNIT=WUNIT
      DO 10 I=1,NELTS
        IF (IMIX.EQ.0.AND.IFLAG(6).EQ.0) THEN
          EDELTA(I)=SFINAL(I)-SINIT1(I)
        ELSE
          EDELTA(I)=SFINAL(I)
        END IF
   10 CONTINUE
      IF (IMIX.EQ.1.OR.IFLAG(6).EQ.1) THEN
```

```
        K=1+IMIX+IFLAG(6)                                               AF 220
        NMINS=NMINS+K                                                   AF 230
        DO 30 I=NMINS,K+1,-1                                            AF 240
          DO 20 J=1,15                                                  AF 250
            PELT(I,J)=PELT(I-K,J)                                       AF 260
   20       PCOEF(I,J)=PCOEF(I-K,J)                                     AF 270
          PNAME(I)=PNAME(I-K)                                           AF 280
          TRANSFER(I)=TRANSFER(I-K)                                     AF 290
   30     FORCE(I)=FORCE(I-K)                                           AF 300
      END IF                                                            AF 310
C                                                                       AF 320
C             CALCULATE COMBINATIONS OF MINERALS AND                    AF 330
C             TEST RESULTING MODELS.                                    AF 340
C                                                                       AF 350
      ITIME=0                                                           AF 360
      MINIGN=100                                                        AF 370
      NUMIGN=0                                                          AF 380
      ISTOP=0                                                           AF 390
   40 MODTOT=0                                                          AF 400
      MODGOOD=0                                                         AF 410
      JFORCE=0                                                          AF 420
      FIRST=.TRUE.                                                      AF 430
      CALL INIT                                                         AF 440
      IF (ITIME.EQ.0) THEN                                              AF 450
        DO 50 I=1,NMINS                                                 AF 460
   50     IF (FORCE(I).EQ.'F') JFORCE=JFORCE+1                          AF 470
        JNUM=1                                                          AF 480
        DO 60 I=1,NMINS-NEQ                                             AF 490
   60     JNUM=JNUM*(NEQ-JFORCE+I)/I                                    AF 500
        IF (JNUM.GT.0) THEN                                             AF 510
          CALL CLS                                                      AF 520
          WRITE (*,70) JNUM                                             AF 530
        END IF                                                          AF 540
      END IF                                                            AF 550
   70 FORMAT (I12,' models to be tested')                               AF 560
C                                                                       AF 570
C             TEST ALL COMBINATIONS                                     AF 580
C                                                                       AF 590
   80 IF (MODTOT.GT.0.AND.MOD(MODTOT,100).EQ.0.AND.ITIME.EQ.0) THEN     AF 600
        IF (MODGOOD.EQ.0.AND.MINIGN.LT.50) THEN                         AF 610
          CALL POSCUR (-4)                                              AF 620
          WRITE (*,*)                                                   AF 630
          CALL CLPART                                                   AF 640
          IF (MINIGN.NE.1) THEN                                         AF 650
            WRITE (*,100) CHAR(27)//'[H',MODTOT,NUMIGN,MINIGN           AF 660
          ELSE                                                          AF 670
            WRITE (*,110) CHAR(27)//'[H',MODTOT,NUMIGN,MINIGN           AF 680
          END IF                                                        AF 690
        ELSE                                                            AF 700
          CALL POSCUR (-4)                                              AF 710
          WRITE (*,*)                                                   AF 720
          CALL CLPART                                                   AF 730
          WRITE (*,90) CHAR(27)//'[H',MODTOT,MODGOOD                    AF 740
        END IF                                                          AF 750
      END IF                                                            AF 760
   90 FORMAT (1X,A,//,I12,' models tested',/,I12,' models found','      AF 770
     *                            ')                                    AF 780
  100 FORMAT (1X,A,//,I12,' models tested',/,I12,' models found',' (igno AF 790
     *ring ',I2,' +/- limitations)')                                    AF 800
  110 FORMAT (1X,A,//,I12,' models tested',/,I12,' models found',' (igno AF 810
     *ring ',I2,' +/- limitation)')                                     AF 820
```

```
      MODTOT=MODTOT+1                                                   AF 830
      CALL NEXT                                                         AF 840
      IF (.NOT.QUIT) GO TO 170                                          AF 850
C                                                                       AF 860
C  DONE WITH ALL THE MODELS (1ST OR 2ND TIME)                           AF 870
C                                                                       AF 880
      IF (ITIME.EQ.1) RETURN                                            AF 890
      MODTOT=MODTOT-1                                                   AF 900
      IGOOD=MODGOOD                                                     AF 910
      IF (NRUN.EQ.1.AND.MODGOOD.EQ.0.AND.NUMIGN.GT.0) THEN              AF 920
        IGOOD=NUMIGN                                                    AF 930
        WRITE (IUNIT,120) CHAR(27)//'[H',MINIGN                         AF 940
      ELSE                                                              AF 950
        MINIGN=0                                                        AF 960
      END IF                                                            AF 970
  120 FORMAT (1X,A,/////' (Ignoring ',I2,' precipitation/dissolution',' AF 980
     *limitations)')                                                    AF 990
      WRITE (IUNIT,130) CHAR(27)//'[H',MODTOT,IGOOD                     AF1000
  130 FORMAT (1X,A,//I12,' models were tested.        '/I12,' models wer AF1010
     *e found which satisfied the constraints.',//)                     AF1020
      IF (NRUN.EQ.0) RETURN                                             AF1030
      ITIME=1                                                           AF1040
      IF (IGOOD.EQ.0) GO TO 240                                         AF1050
      IF (IGOOD.EQ.1) THEN                                              AF1060
        ISTOP=1                                                         AF1070
        GO TO 40                                                        AF1080
      END IF                                                            AF1090
  140 WRITE (*,150)                                                     AF1100
  150 FORMAT (/,' Display models: <A>ll at once, [ENTER] for',' each, or AF1110
     * <N>one.')                                                        AF1120
      READ (*,160) ANS                                                  AF1130
      WRITE (IUNIT,'(1X)')                                              AF1140
  160 FORMAT (A)                                                        AF1150
      ANS=UPCS(ANS)                                                     AF1160
      IF (ANS.NE.'A'.AND.ANS.NE.'N'.AND.ANS.NE.' ') GO TO 140           AF1170
      IF (ANS.EQ.'N') RETURN                                            AF1180
      IF (ANS.EQ.' ') ISTOP=1                                           AF1190
      GO TO 40                                                          AF1200
  170 CALL SET                                                          AF1210
      CALL ROWCOL                                                       AF1220
      IF (.NOT.ARRAYOK) GO TO 80                                        AF1230
      CALL SLNQ (NEQ,A,DELTA,SINGULAR)                                  AF1240
      IF (SINGULAR) GO TO 80                                            AF1250
      MODELOK=.TRUE.                                                    AF1260
      TEST='TRUE'                                                       AF1270
      IBAD=0                                                            AF1280
      DO 180 I=1,NEQ                                                    AF1290
        K=NOW(I)                                                        AF1300
        IF (TRANSFER(K).EQ.'-'.AND.DELTA(I).GT.0.) IBAD=IBAD+1          AF1310
        IF (TRANSFER(K).EQ.'+'.AND.DELTA(I).LT.0.) IBAD=IBAD+1          AF1320
        IF (IFLAG(1).EQ.1.AND.I.EQ.2.AND.IBAD.GT.0) TEST='FALSE'        AF1330
        IF (DABS(DELTA(I)).GT.1D5) TEST='FALSE'                         AF1340
        IF (TEST.EQ.'FALSE') MODELOK=.FALSE.                            AF1350
  180 CONTINUE                                                          AF1360
      IF (MODELOK.EQV..FALSE.) GO TO 80                                 AF1370
      IF (IBAD.GT.0.AND.IBAD.LT.MINIGN) THEN                            AF1380
        MINIGN=IBAD                                                     AF1390
        NUMIGN=1                                                        AF1400
        GO TO 80                                                        AF1410
      ELSE IF (IBAD.EQ.MINIGN.AND.ITIME.EQ.0) THEN                      AF1420
        NUMIGN=NUMIGN+1                                                 AF1430
```

```
        GO TO 80
      ELSE IF (IBAD.GT.MINIGN) THEN
        GO TO 80
      END IF
C
C  WE'VE GOT A GOOD ONE (POSSIBLY IGNORING CONSTRAINTS)
C
      MODGOOD=MODGOOD+1
      IF (ITIME.EQ.0.AND.NRUN.EQ.1) GO TO 80
      CALL PRINT
      IF (NRUN.EQ.0) THEN
        CALL CISO (2)
        CALL CISO (1)
        GO TO 80
      END IF
      IF (MODGOOD.EQ.IGOOD.AND.ISTOP.EQ.0) GO TO 240
      IF (MODGOOD.EQ.IGOOD.AND.ISTOP.EQ.1) GO TO 220
      IF (ISTOP.EQ.0) GO TO 80
  190 IF (IFLAG(3).EQ.0) THEN
        WRITE (*,210)
      ELSE
        WRITE (*,200)
      END IF
  200 FORMAT (' Hit <S> for Rayleigh data, <C> to run all C-14 models, '
     *,' <ENTER> to continue,'/' or any other key to quit.')
      READ (*,160) ANS
      IF (UPCS(ANS).EQ.'S'.AND.IFLAG(3).EQ.1) THEN
        CALL CISO (1)
        GO TO 190
      END IF
      IF (UPCS(ANS).EQ.'C'.AND.IFLAG(3).EQ.1) THEN
        CALL CISO (2)
        GO TO 190
      END IF
  210 FORMAT (' <Enter> to continue, or any other key to quit')
      IF (ANS.NE.' ') RETURN
      GO TO 80
  220 IF (IFLAG(3).EQ.1) THEN
        WRITE (*,230)
      ELSE
        WRITE (*,250)
      END IF
  230 FORMAT (//,' No more models, <S> to show Rayleigh data',', <C> to
     *run all C-14 models,'/' <Enter> to continue')
      READ (*,160) ANS
      IF (IFLAG(3).EQ.0) RETURN
      IF (UPCS(ANS).EQ.'S') THEN
        CALL CISO (1)
      ELSE IF (UPCS(ANS).EQ.'C') THEN
        CALL CISO (2)
      ELSE
        RETURN
      END IF
      GO TO 220
  240 WRITE (*,250)
  250 FORMAT (//,' No more models, hit <Enter> to continue')
      READ (*,160) ANS
      RETURN
      END
      DOUBLE PRECISION FUNCTION C14 (IWHICH,IWELL)
C
```

```
C The result of a specific A0 model for the selected initial well
C is returned. The convention adopted is that total dissolved carbon
C is the sum of total dissolved inorganic carbon, dissolved methane,
C and dissolved organic carbon.
C
$INCLUDE:'NETPATH.BLO'
      I=WELL(IWELL)
C
C C1 is the value used for initial well C-13
C
      C1=DBDATA(I,21)/DBDATA(I,41)
      IF (I11.GT.0.) C1=C14DAT(3)
C
C C2 is the value for C-13 of soil gas CO2
C
      C2=C14DAT(5)
C
C Define C-13 of soil gas CO2 by mass balance:
C Pure Water + Calcite + Dolomite + Gypsum + CO2 (without fractionation)
C     = Initial Well
C This case is comparable to piston injection of CO2 with subsequent
C closed-system reaction with calcite, dolomite and gypsum
C
      IF (I10.GE.1.) C2=(DBDATA(I,21)-2*DBDATA(I,5)*C14DAT(12)-
     *(DBDATA(I,3)-DBDATA(I,5)-DBDATA(I,30))*C14DAT(13))/(DBDATA(I,41)+
     *DBDATA(I,30)-DBDATA(I,3)-DBDATA(I,5))
C
C Define C-13 of soil gas CO2 for open (equilibrium) system
C
      CALL CFRACT (FRACTION,2,2D0-IWELL,IWELL-1D0,IERROR)
      IF (I10.EQ.3.) C2=C1+FRACTION
C
C Define C-13 of soil gas CO2 for partial open system
C This case is analogous to an open equilibrium system in a quartz
C sand aquifer with subsequent reaction with calcite, dolomite and
C gypsum in a closed system.
C
      CALL CFRACT (FRACTION,3,2D0-IWELL,IWELL-1D0,IERROR)
      IF (I10.EQ.2.) C2=C2-FRACTION
C
C RETURNS C-13 OF SOIL CO2 VALUE FOR CHECKING...
      IF (IWHICH.EQ.-1) THEN
        C14=C2
        RETURN
      END IF
C
      C3=C14DAT(2)+.2*FRACTION
      C4=C2+FRACTION
      IF (DBDATA(I,1).LE.0..AND.(IWHICH.NE.9.AND.IWHICH.NE.3)) GO TO 10
      GO TO (20,50,40,60,70,80,90,100,30),IWHICH
   10 C14=0
      RETURN
C Original Data
   20 C14=(DBDATA(I,22)+DBDATA(I,42)*DBDATA(I,46)+DBDATA(I,43)*DBDATA(I,
     *47))/DBDATA(I,1)
      RETURN
C User-entered value
   30 C14=C14DAT(5+IWELL)
      RETURN
C Vogel model
   40 C14=(85.*DBDATA(I,41)+DBDATA(I,42)*DBDATA(I,46)+DBDATA(I,43)*
```

```
     *DBDATA(I,47))/DBDATA(I,1)
      RETURN
C Mass Balance model (calcite, dolomite, gypsum, co2 gas)
   50 C14=((DBDATA(I,5)*C14DAT(8)+C14DAT(9)*(DBDATA(I,3)-DBDATA(I,5)-
     *DBDATA(I,30))+C14DAT(2)*(DBDATA(I,1)+DBDATA(I,30)-DBDATA(I,5)-
     *DBDATA(I,3)))+DBDATA(I,42)*DBDATA(I,46)+DBDATA(I,43)*DBDATA(I,47))
     */DBDATA(I,1)
      RETURN
C Tamers
   60 C14=((((DBDATA(I,38)+.5*DBDATA(I,36))*C14DAT(2)+C14DAT(1)*.5*
     *DBDATA(I,36))/(DBDATA(I,38)+DBDATA(I,36)))*DBDATA(I,41)+DBDATA(I,
     *42)*DBDATA(I,46)+DBDATA(I,43)*DBDATA(I,47))/DBDATA(I,1)
      RETURN
C Ingerson and Pearson
   70 IF (C14DAT(4).EQ.C2) THEN
        C14=0.
        RETURN
      END IF
      C14=(((C1-C14DAT(4))/(C2-C14DAT(4))*(C14DAT(2)-C14DAT(1))+
     *C14DAT(1))*DBDATA(I,41)+DBDATA(I,42)*DBDATA(I,46)+DBDATA(I,43)*
     *DBDATA(I,47))/DBDATA(I,1)
      RETURN
C Mook
   80 AH=10**(-DBDATA(I,37))
      T=DBDATA(I,25)+273.15
      AK1=10**(-3404.71/T-.032786*T+14.8435)
      AK2=10**(-2902.39/T-0.02379*T+6.4890)
      AA=AH*AH/(AH*AH+AH*AK1+AK1*AK2)
      B=AH*AK1/(AH*AH+AH*AK1+AK1*AK2)
      CALL CFRACT (FRACTION,4,2D0-IWELL,IWELL-1D0,IERROR)
      C14=(AA+.5*B)*C3+.5*B*C14DAT(1)+(C14DAT(2)*(1.-FRACTION/500.)-0.5*
     *(C3+C14DAT(1)))*(C1-AA*C4-.5*B*(C4+C14DAT(4)))/(C2-FRACTION*(1.+
     *C2/1000.)-.5*(C4+C14DAT(4)))
      C14=(C14*DBDATA(I,41)+DBDATA(I,42)*DBDATA(I,46)+DBDATA(I,43)*
     *DBDATA(I,47))/DBDATA(I,1)
      RETURN
C Fontes and Garnier
   90 CM=DBDATA(I,39)+DBDATA(I,36)/2
      CALL CFRACT (FRACTION,5,2D0-IWELL,IWELL-1D0,IERROR)
      C14=(1-CM/DBDATA(I,41))*C14DAT(2)+CM/DBDATA(I,41)*C14DAT(1)+
     *(C14DAT(2)-.2*FRACTION-C14DAT(1))*(C1-CM/DBDATA(I,41)*C14DAT(4)-
     *(1-CM/DBDATA(I,41))*C2)/(C2-FRACTION-C14DAT(4))
      C14=(C14*DBDATA(I,41)+DBDATA(I,42)*DBDATA(I,46)+DBDATA(I,43)*
     *DBDATA(I,47))/DBDATA(I,1)
      RETURN
C Eichinger
  100 CALL CFRACT (FRACT9,7,2D0-IWELL,IWELL-1D0,IERROR)
      CALL CFRACT (FRACT8,6,2D0-IWELL,IWELL-1D0,IERROR)
      CALL CFRACT (FRACT5,3,2D0-IWELL,IWELL-1D0,IERROR)
C     Eqn. (6) of Eichinger is corrected (Fontes, written comm., 1991)
      C13SEQ=DBDATA(I,38)/DBDATA(I,41)*(FRACT9)-DBDATA(I,36)/DBDATA(I,
     *41)*FRACT8+C14DAT(4)
      C13BEX=DBDATA(I,38)/DBDATA(I,41)*(C2+FRACT5)+0.5*DBDATA(I,36)/
     *DBDATA(I,41)*(C2+FRACT5+C14DAT(4))
      C14=(C1-C13SEQ)/(C13BEX-C13SEQ)*(DBDATA(I,38)+.5*DBDATA(I,36))/
     *DBDATA(I,41)*C14DAT(2)
      C14=(C14*DBDATA(I,41)+DBDATA(I,42)*DBDATA(I,46)+DBDATA(I,43)*
     *DBDATA(I,47))/DBDATA(I,1)
      RETURN
      END
      SUBROUTINE CFRACT (FRACTION,ITIME,FRAC1,FRAC2,IERROR)
```

```
C
C Fractionation factors for the inorganic carbon species are calculated
C from the equations of Mook, and Deines et al.  Calculations are as
C alpha, and the additive fractionation factor, Epsilon, is returned.
C
C A specific frationation factor is calculated, based on some fraction
C of the initial and final waters.  FRAC1 and FRAC2 are the fractions of
C the two initial wells to use, if applicable.
C
C Meaning of ITIME 1:  Calcite-solution at some point along the flowpath
C                  2:  CO2(g)-solution at some point along the flowpath
C                  3:  CO2(aq)-CO2(g) at the initial well (Eichinger)
C                  4:  CO2(g)-HCO3 at the initial well (Mook)
C                  5:  CO2(g)-calcite at the initial well
C                      (Eqn. as in Fontes and Garnier)
C                  6:  Calcite-HCO3 at the initial well (Eichinger)
C                  7:  CO2(aq)-Calcite at the initial well (Eichinger)
C
$INCLUDE:'NETPATH.BLO'
      IERROR=0
      K=1
C
C  SET UP CARBON AND TEMP DATA
C
   10 T=DBDATA(WELL(K),25)+273.15
      H2CO3=DBDATA(WELL(K),38)
      HCO3=DBDATA(WELL(K),36)
      CO3=DBDATA(WELL(K),39)
      CH4=DBDATA(WELL(K),42)
      DOC=DBDATA(WELL(K),43)
      DCO2=0
      IF (DBDATA(WELL(K),41).NE.0) DCO2=DBDATA(WELL(K),21)/
     *DBDATA(WELL(K),41)
      DCH4=DBDATA(WELL(K),44)
      DDOC=DBDATA(WELL(K),45)
      DC=(DBDATA(WELL(K),21)+DBDATA(WELL(K),42)*DBDATA(WELL(K),44)+
     *DBDATA(WELL(K),43)*DBDATA(WELL(K),45))/DBDATA(WELL(K),1)
C
C HANDLE MIXING
C
      IF (K.EQ.1.AND.IFLAG(1).EQ.1) THEN
        T=T*FRAC1+(DBDATA(WELL(2),25)+273.15)*FRAC2
        H2CO3=H2CO3*FRAC1+FRAC2*DBDATA(WELL(2),38)
        HCO3=HCO3*FRAC1+FRAC2*DBDATA(WELL(2),36)
        CO3=CO3*FRAC1+FRAC2*DBDATA(WELL(2),39)
        IF (DBDATA(WELL(1),41).NE.0) THEN
          IF (DBDATA(WELL(2),41).NE.0) THEN
            DCO2=DBDATA(WELL(1),21)*FRAC1+FRAC2*DBDATA(WELL(2),21)
            DCO2=DCO2/(DBDATA(WELL(1),41)*FRAC1+FRAC2*DBDATA(WELL(2),41)
     *       )
          ELSE
            DCO2=DBDATA(WELL(1),21)/DBDATA(WELL(1),41)*FRAC1
          END IF
        ELSE
          IF (DBDATA(WELL(2),41).NE.0) THEN
            DCO2=DBDATA(WELL(2),21)/DBDATA(WELL(2),41)*FRAC2
          ELSE
```

```
            DCO2=0                                                      AH 590
          END IF                                                        AH 600
        END IF                                                          AH 610
        DC=((DBDATA(WELL(1),21)+DBDATA(WELL(1),42)*DBDATA(WELL(1),44)+  AH 620
     *   DBDATA(WELL(1),43)*DBDATA(WELL(1),45))*FRAC1+(DBDATA(WELL(2),  AH 630
     *   21)+DBDATA(WELL(2),42)*DBDATA(WELL(2),44)+DBDATA(WELL(2),43)*  AH 640
     *   DBDATA(WELL(2),45))*FRAC2)/(DBDATA(WELL(1),1)*FRAC1+           AH 650
     *   DBDATA(WELL(2),1)*FRAC2)                                       AH 660
      END IF                                                            AH 670
C                                                                       AH 680
C Calculate fractionation factors (alpha)                               AH 690
C                                                                       AH 700
      IF (IFLAG(5).EQ.0) THEN                                           AH 710
C                   Mook (1980) numbers                                 AH 720
C CO2(aq) - HCO3                                                        AH 730
        EBA=(24.12-9866/T)/1000+1                                       AH 740
C CO3 - HCO3                                                            AH 750
        EBC=(1.63-604/T)/1000+1                                         AH 760
C Calcite - HCO3                                                        AH 770
        EBCAL=(15.10-4232/T)/1000+1                                     AH 780
C CO2(g) - HCO3                                                         AH 790
        EBGAS=(23.89-9483/T)/1000+1                                     AH 800
      ELSE                                                              AH 810
C                  Deines, et al. (1974) numbers                        AH 820
C CO2(aq) - CO2(g)                                                      AH 830
        DK0=DEXP((-0.91+6300./(T*T))/1000.)                             AH 840
C HCO3 - CO2(g)                                                         AH 850
        DK1=DEXP((-4.54+1099000./(T*T))/1000.)                          AH 860
C CO3 - CO2(g)                                                          AH 870
        DK2=DEXP((-3.4+870000./(T*T))/1000.)                            AH 880
C Calcite - CO2(g)                                                      AH 890
        DK3=DEXP((-3.63+1194000./(T*T))/1000.)                          AH 900
C CO2(aq) - HCO3                                                        AH 910
        EBA=DK0/DK1                                                     AH 920
C CO3 - HCO3                                                            AH 930
        EBC=DK2/DK1                                                     AH 940
C Calcite - HCO3                                                        AH 950
        EBCAL=DK3/DK1                                                   AH 960
C CO2(g) - HCO3                                                         AH 970
        EBGAS=1./DK1                                                    AH 980
      END IF                                                            AH 990
C CO2(aq) - CO2(g) (for either one)                                     AH1000
      IF (ITIME.EQ.3) FRACTION=EBA/EBGAS                                AH1010
      IF (ITIME.EQ.7) FRACTION=EBA/EBCAL                                AH1020
      IF (ITIME.EQ.6) FRACTION=EBCAL                                    AH1030
      IF (ITIME.EQ.5) FRACTION=((T-273.15)*.1-12.38)/1000+1             AH1040
      EB=EBCAL                                                          AH1050
      IF (ITIME.EQ.2) EB=EBGAS                                          AH1060
C     IF (ITIME.EQ.4) FRACTION=(23.89-9483/T)/1000+1                    AH1070
      IF (ITIME.GT.2) FRACTION=(FRACTION-1D0)*1D3                       AH1080
      IF (ITIME.GT.2) RETURN                                            AH1090
      DIST13=EBA*H2CO3+HCO3+EBC*CO3                                     AH1100
      TDIC=H2CO3+HCO3+CO3                                               AH1110
      CTOT=TDIC+CH4+DOC                                                 AH1120
      DEL13=TDIC*(1000+DCO2)/DIST13-1D3                                 AH1130
C KINETIC ALPHA DOC - HCO3                                              AH1140
      ADOC=(1000+DDOC)/(1000+DEL13)                                     AH1150
C KINETIC ALPHA CH4 - HCO3                                              AH1160
      ACH4=(1000+DCH4)/(1000+DEL13)                                     AH1170
      DIST13=CTOT/(DIST13+CH4*ACH4+DOC*ADOC)                            AH1180
      IF (K.EQ.1) THEN                                                  AH1190
```

```
        FRACTION=EB*DIST13
        FRACTION=(FRACTION-1D0)*1D3
        K=K+2
        GO TO 10
      END IF
      FRACTION=(1.-DISALONG)*FRACTION+DISALONG*((EB*DIST13)-1D0)*1D3
      RETURN
      END
      SUBROUTINE CISO (ISCR)
C
C The isotopic calculations, according to the Rayleigh equations, are
C carried out.  If there is any reason why they cannot be carried out, a
C warning message is printed.  Depending on ISCR, different values are
C computed and different data are displayed.  To help locate problems,
C values of variables in the Rayleigh equation are stored in NETPATH.OUT
C
$INCLUDE:'NETPATH.BLO'
      CHARACTER*8 DISPHA(39),PREPHA(39)
      INTEGER ELMT(4),ISOT(4)
      LOGICAL DOWEHAVE
      DATA ELMT/1,1,2,15/,ISOT/21,22,23,24/
      IF (ISCR.GE.1) GO TO 20
      OPEN (9,FILE='NETPATH.OUT')
      WRITE (IUNIT,10)
      IAGE=0
   10 FORMAT (/,15X,' Computed    Observed')
   20 DO 330 I=1,4
        IF (I.NE.2.AND.ISCR.EQ.2) GO TO 330
        IF (ISCR.EQ.1) WRITE (IUNIT,30) ELELONG(ISOT(I))
   30   FORMAT (/,' Data used for ',A)
        IERR=0
        IF (NODATA(WELL(1),ISOT(I)).EQ.1.OR.NODATA(WELL(3),ISOT(I)).EQ.
     *   1) GO TO 190
        IF (IFLAG(1).EQ.1) THEN
          IF (NODATA(WELL(2),ISOT(I)).EQ.1) GO TO 190
        END IF
        CINIT=DBDATA(WELL(1),ELMT(I))
        IF (CINIT.LE.0) GO TO 190
        DINIT=DBDATA(WELL(1),ISOT(I))/CINIT
        IF (I.EQ.1) THEN
          DINIT=DBDATA(WELL(1),ISOT(I))
          DINIT=DINIT+DBDATA(WELL(1),ISOT(I)*2+2)*DBDATA(WELL(1),42)
          DINIT=DINIT+DBDATA(WELL(1),ISOT(I)*2+3)*DBDATA(WELL(1),43)
          DINIT=DINIT/CINIT
        END IF
        IF (IFLAG(1).EQ.1) THEN
          CINIT=DELTA(1)*DBDATA(WELL(1),ELMT(I))+DELTA(2)*DBDATA(WELL(2)
     *      ,ELMT(I))
          DINIT=(DELTA(1)*DBDATA(WELL(1),ISOT(I))+DELTA(2)*
     *      DBDATA(WELL(2),ISOT(I)))/CINIT
          IF (I.EQ.1) THEN
            DINIT=DBDATA(WELL(1),ISOT(I))*DELTA(1)+DBDATA(WELL(2),
     *        ISOT(I))*DELTA(2)
            DINIT=DINIT+DBDATA(WELL(1),ISOT(I)*2+2)*DBDATA(WELL(1),42)*
     *        DELTA(1)+DBDATA(WELL(2),ISOT(I)*2+2)*DBDATA(WELL(2),42)*
     *        DELTA(2)
            DINIT=DINIT+DBDATA(WELL(1),ISOT(I)*2+3)*DBDATA(WELL(1),43)*
     *        DELTA(1)+DBDATA(WELL(2),ISOT(I)*2+3)*DBDATA(WELL(2),43)*
     *        DELTA(2)
            DINIT=DINIT/CINIT
          END IF
```

```
      END IF                                                            AI 540
      IF (I.EQ.2) THEN                                                  AI 550
        IF (IFLAG(1).EQ.0) THEN                                         AI 560
          DINIT=C14(IFLAG(4),1)                                         AI 570
        ELSE                                                            AI 580
          DINIT=C14(IFLAG(4),1)*DELTA(1)+DELTA(2)*C14(IFLAG(4),2)       AI 590
        END IF                                                          AI 600
        DINIT=10.*DINIT-1000.                                           AI 610
      END IF                                                            AI 620
      IF (ISCR.EQ.1) THEN                                               AI 630
        IF (I.EQ.2) THEN                                                AI 640
          WRITE (IUNIT,40) DINIT/10.+100.                               AI 650
        ELSE                                                            AI 660
          WRITE (IUNIT,40) DINIT                                        AI 670
        END IF                                                          AI 680
      END IF                                                            AI 690
 40   FORMAT (' Initial Value:',F12.4)                                  AI 700
      CFINAL=DBDATA(WELL(3),ELMT(I))                                    AI 710
      IPRE=0                                                            AI 720
      IDIS=0                                                            AI 730
      IDELT=0                                                           AI 740
 50   IDELT=IDELT+1                                                     AI 750
      IF (IDELT.GT.NEQ) GO TO 90                                        AI 760
      IPHA=0                                                            AI 770
 60   IPHA=IPHA+1                                                       AI 780
      IF (IPHA.GT.NOPHA) GO TO 50                                       AI 790
      IF (PNAME(NOW(IDELT)).EQ.PHASE(IPHA)) GO TO 70                    AI 800
      GO TO 60                                                          AI 810
 70   CIC=0                                                             AI 820
      DO 80 J=1,14                                                      AI 830
 80     IF (JELE(IPHA,J).EQ.ELMT(I)) CIC=PCOEFF(IPHA,J)                 AI 840
      IF (CIC.EQ.0) GO TO 50                                            AI 850
      IF (DELTA(IDELT).GT.0.0.OR.IT(IPHA).EQ.'*') THEN                  AI 860
        IDIS=IDIS+1                                                     AI 870
        DISPHA(IDIS)=PHASE(IPHA)                                        AI 880
        DISDAT(IDIS,1)=0.0                                              AI 890
        IF (DELTA(IDELT).GT.0.0) DISDAT(IDIS,1)=DELTA(IDELT)*CIC        AI 900
        IF (IT(IPHA).EQ.'*') DISDAT(IDIS,1)=DISDAT(IDIS,1)+PARA(IPHA,   AI 910
     *   1)*CIC                                                         AI 920
        DISDAT(IDIS,2)=PARA(IPHA,I+1)                                   AI 930
        IF (.NOT.DOWEHAVE(IPHA,I+1)) IERR=1                             AI 940
        IF (PHASE(IPHA).EQ.'CO2-CH4 ') THEN                             AI 950
          DISDAT(IDIS,2)=PARA(IPHA,I+1)*P(1)+(1.-P(1))*PARA(IPHA,I+3)   AI 960
          IF (DOWEHAVE(IPHA,I+3)) THEN                                  AI 970
            IF (P(1).EQ.0.) IERR=0                                      AI 980
          ELSE                                                          AI 990
            IF (P(1).LT.1.) IERR=1                                      AI1000
          END IF                                                        AI1010
        END IF                                                          AI1020
        IF (I.EQ.2) DISDAT(IDIS,2)=DISDAT(IDIS,2)*10.-1000.             AI1030
      END IF                                                            AI1040
      IF (DELTA(IDELT).LT.0.0.OR.IT(IPHA).EQ.'*') THEN                  AI1050
        IPRE=IPRE+1                                                     AI1060
        PREPHA(IPRE)=PNAME(NOW(IDELT))                                  AI1070
        PREDAT(IPRE,1)=0.0                                              AI1080
        IF (DELTA(IDELT).LT.0.0) PREDAT(IPRE,1)=-DELTA(IDELT)*CIC       AI1090
        IF (IT(IPHA).EQ.'*') PREDAT(IPRE,1)=PREDAT(IPRE,1)+PARA(IPHA,   AI1100
     *   1)*CIC                                                         AI1110
        PREDAT(IPRE,2)=PARA(IPHA,I+5)                                   AI1120
        IF (.NOT.DOWEHAVE(IPHA,I+5)) IERR=1                             AI1130
        IF (PHASE(IPHA).EQ.'CO2-CH4 ') THEN                             AI1140
```

```
            IF (.NOT.DOWEHAVE(IPHA,7+I).AND.P(1).LT.1.) THEN            AI1150
              IERR=1                                                    AI1160
            ELSE                                                        AI1170
              IERR=0                                                    AI1180
              IF (DOWEHAVE(IPHA,5+I)) THEN                              AI1190
                PREDAT(IPRE,2)=PARA(IPHA,5+I)*P(1)+(1.-P(1))*PARA(IPHA, AI1200
     *           7+I)                                                   AI1210
              ELSE                                                      AI1220
                CALL CFRACT (FRACTION,2,DELTA(1),DELTA(2),IERROR)       AI1230
                PREDAT(IPRE,2)=I*FRACTION*P(1)+(1.-P(1))*PARA(IPHA,7+I) AI1240
              END IF                                                    AI1250
            END IF                                                      AI1260
          ELSE IF (PHASE(IPHA)(1:3).EQ.'CO2'.AND.I.LT.3) THEN           AI1270
            IERR=0                                                      AI1280
            CALL CFRACT (FRACTION,2,DELTA(1),DELTA(2),IERROR)           AI1290
            IF (.NOT.DOWEHAVE(IPHA,I+5)) PREDAT(IPRE,2)=I*FRACTION      AI1300
          ELSE IF (I.LT.3.AND.ISTATE(IPHA,1).EQ.4.AND..NOT.             AI1310
     *     DOWEHAVE(IPHA,I+5)) THEN                                     AI1320
            IERR=0                                                      AI1330
            CALL CFRACT (FRACTION,1,DELTA(1),DELTA(2),IERROR)           AI1340
            PREDAT(IPRE,2)=I*FRACTION                                   AI1350
          ELSE IF (I.EQ.3.AND.(ISTATE(IPHA,2).EQ.-1.OR.ISTATE(IPHA,2).  AI1360
     *     EQ.-2)) THEN                                                 AI1370
            CALL SFRACT (FRACTION,1,DELTA(1),DELTA(2),IERROR)           AI1380
            IF (IERROR.EQ.1) FRACTION=0.                                AI1390
            PREDAT(IPRE,2)=PREDAT(IPRE,2)+FRACTION                      AI1400
          ELSE IF (I.EQ.3.AND.ISTATE(IPHA,2).EQ.6) THEN                 AI1410
            CALL SFRACT (FRACTION,2,DELTA(1),DELTA(2),IERROR)           AI1420
            IF (IERROR.EQ.1) FRACTION=0.                                AI1430
            PREDAT(IPRE,2)=PREDAT(IPRE,2)+FRACTION                      AI1440
          END IF                                                        AI1450
        END IF                                                          AI1460
        IF (IERR.EQ.1) GO TO 190                                        AI1470
        GO TO 50                                                        AI1480
   90   IF (ISCR.EQ.1) GO TO 220                                        AI1490
        IF (ISCR.EQ.2) GO TO 290                                        AI1500
        WRITE (9,100) ELELONG(ISOT(I))                                  AI1510
  100   FORMAT (1X,A)                                                   AI1520
        WRITE (9,130) IDIS                                              AI1530
        DO 110 JJ=1,IDIS                                                AI1540
  110     WRITE (9,150) DISDAT(JJ,1),DISDAT(JJ,2)                       AI1550
        WRITE (9,140) IPRE                                              AI1560
        DO 120 JJ=1,IPRE                                                AI1570
  120     WRITE (9,150) PREDAT(JJ,1),PREDAT(JJ,2)                       AI1580
  130   FORMAT (I3,' dissolving:')                                      AI1590
  140   FORMAT (I3,' precipitating:')                                   AI1600
  150   FORMAT (1X,2(F15.6))                                            AI1610
        CALL RAYLEIGH (0)                                               AI1620
        IF (I.EQ.2) RESULT=(1000.+RESULT)/10.                           AI1630
        DFINAL=DBDATA(WELL(3),ISOT(I))/DBDATA(WELL(3),ELMT(I))          AI1640
        IF (ELMT(I).EQ.1) DFINAL=(DBDATA(WELL(3),ISOT(I))+DBDATA(WELL(3) AI1650
     *   ,42)*DBDATA(WELL(3),ISOT(I)*2+2)+DBDATA(WELL(3),43)*           AI1660
     *   DBDATA(WELL(3),ISOT(I)*2+3))/DBDATA(WELL(3),1)                 AI1670
        IF (I.EQ.2.AND.DFINAL*RESULT.GT.0.0) AGE=5730./DLOG(2D0)*       AI1680
     *   DLOG(RESULT/DFINAL)                                            AI1690
        IF (I.EQ.2.AND.DFINAL*RESULT.GT.0.001.AND.RESULT.GT.0.001) IAGE= AI1700
     *   1                                                              AI1710
        IF (I.EQ.2) THEN                                                AI1720
          IF (IAGE.EQ.1) THEN                                           AI1730
            WRITE (IUNIT,170) ELELONG(ISOT(I)),RESULT,DFINAL            AI1740
          ELSE                                                          AI1750
```

```
          WRITE (IUNIT,160) ELELONG(ISOT(I)),RESULT,DFINAL              AI1760
        END IF                                                          AI1770
      ELSE IF (I.EQ.4) THEN                                             AI1780
        WRITE (IUNIT,180) ELELONG(ISOT(I)),RESULT,DFINAL                AI1790
      ELSE                                                              AI1800
        WRITE (IUNIT,160) ELELONG(ISOT(I)),RESULT,DFINAL                AI1810
        JHAVE=0                                                         AI1820
        IF (DBDATA(WELL(1),42).NE.0.OR.DBDATA(WELL(1),43).NE.0) JHAVE=  AI1830
   *     1                                                              AI1840
        IF (DBDATA(WELL(3),42).NE.0.OR.DBDATA(WELL(3),43).NE.0) JHAVE=  AI1850
   *     1                                                              AI1860
        IF (IFLAG(1).EQ.1) THEN                                         AI1870
          IF (DBDATA(WELL(2),42).NE.0.OR.DBDATA(WELL(2),43).NE.0)       AI1880
   *       JHAVE=1                                                      AI1890
        END IF                                                          AI1900
        IF (I.EQ.1.AND.DBDATA(WELL(3),41).NE.0.AND.JHAVE.EQ.1)          AI1910
   *     WRITE (IUNIT,160) 'DIC C-13     ',(RESULT*CFINAL-              AI1920
   *     DBDATA(WELL(3),42)*DBDATA(WELL(3),44)-DBDATA(WELL(3),43)*      AI1930
   *     DBDATA(WELL(3),45))/DBDATA(WELL(3),41),DBDATA(WELL(3),21)/     AI1940
   *     DBDATA(WELL(3),41)                                             AI1950
      END IF                                                            AI1960
      GO TO 330                                                         AI1970
160   FORMAT (1X,A,' ',F10.4,2X,F10.4)                                  AI1980
170   FORMAT (1X,A,' ',F10.4,'*',1X,F10.4)                              AI1990
180   FORMAT (1X,A,' ',F10.6,2X,F10.6)                                  AI2000
190   IF (ISCR.EQ.0) WRITE (IUNIT,200) ELELONG(ISOT(I))                 AI2010
      IF (ISCR.EQ.1) WRITE (IUNIT,210)                                  AI2020
200   FORMAT (1X,A,6X,'Insufficient data')                              AI2030
210   FORMAT (7X,'Insufficient data')                                   AI2040
      GO TO 330                                                         AI2050
220   IF (IPRE+IDIS.EQ.0) THEN                                          AI2060
        WRITE (IUNIT,'(A)') ' No incoming or outgoing phases'           AI2070
        GO TO 330                                                       AI2080
      END IF                                                            AI2090
      IF (IDIS.GT.0) WRITE (IUNIT,230) IDIS                             AI2100
230   FORMAT (I3,' dissolving:')                                        AI2110
      DO 240 J=1,IDIS                                                   AI2120
        IF (I.EQ.2) THEN                                                AI2130
          WRITE (IUNIT,250) DISPHA(J),DISDAT(J,1),(DISDAT(J,2)+1000.)/  AI2140
   *       10.                                                          AI2150
        ELSE                                                            AI2160
          WRITE (IUNIT,250) DISPHA(J),DISDAT(J,1),DISDAT(J,2)           AI2170
        END IF                                                          AI2180
240   CONTINUE                                                          AI2190
250   FORMAT (5X,A8,2X,'Delta element:',F10.5,'  Isotopic composition:  AI2200
   *',F10.4)                                                            AI2210
      IF (IPRE.GT.0) WRITE (IUNIT,260) IPRE                             AI2220
260   FORMAT (I3,' precipitating:')                                     AI2230
      DO 270 J=1,IPRE                                                   AI2240
270     WRITE (IUNIT,280) PREPHA(J),PREDAT(J,1),PREDAT(J,2)             AI2250
280   FORMAT (5X,A8,2X,'Delta element:',F10.5,'  Fractionation factor:  AI2260
   *',F10.4)                                                            AI2270
      GO TO 330                                                         AI2280
290   IAGE=0                                                            AI2290
      WRITE (IUNIT,300)                                                 AI2300
300   FORMAT ('  Model                    A0     Computed  Observed','  AI2310
   *     Age',/,33X,'(no decay)',/,1X,60('-'))                          AI2320
      DO 320 IMOD=1,9                                                   AI2330
        IF (IFLAG(1).EQ.0) THEN                                         AI2340
          DINIT=C14(IMOD,1)                                             AI2350
        ELSE                                                            AI2360
```

```
            DINIT=C14(IMOD,1)*DELTA(1)+DELTA(2)*C14(IMOD,2)
          END IF
          IF (DINIT.LE.0) GO TO 320
          A0=DINIT
          DINIT=DINIT*10-1000.
          CALL RAYLEIGH (1)
          RESULT=(1000.+RESULT)/10.
          DFINAL=DBDATA(WELL(3),ISOT(I))/DBDATA(WELL(3),ELMT(I))
          IF (DFINAL*RESULT.LE.0.00001) GO TO 320
          AGE=5730./DLOG(2D0)*DLOG(RESULT/DFINAL)
          WRITE (IUNIT,310) MODEL(IMOD),A0,RESULT,DFINAL,AGE
  310     FORMAT (1X,A,3(F10.2),F10.0)
  320   CONTINUE
  330 CONTINUE
      IF (IAGE.EQ.1) WRITE (IUNIT,340) AGE,MODEL(IFLAG(4))
  340 FORMAT (1X,73('-'),/,' Adjusted C-14 age in years: ',F7.0,'*',5X,'
     ** = Based on ',A)
      WRITE (IUNIT,*)
      CLOSE (9)
      RETURN
      END
      SUBROUTINE CLPART
C
C The screen is cleared from the cursor 3 lines down. Works for VT100
C
      WRITE (*,*) CHAR(27)//'[B'//CHAR(27)//'[B'//CHAR(27)//'[K'//
     *CHAR(27)//'[A'//CHAR(27)//'[K'//CHAR(27)//'[A'//CHAR(27)//'[K'//
     *CHAR(27)//'[A'
      RETURN
      END
      SUBROUTINE CLS
C
C The screen is cleared.  Works for VT100
C
      WRITE (*,*) CHAR(27)//'[2J'
      RETURN
      END
      SUBROUTINE CONLIST (LAST)
C
C All the constraints considered by the program are displayed, up to
C LAST.  Compare this with LISTCON which lists only those constraints
C currently under consideration.
C
$INCLUDE:'NETPATH.BLO'
      WRITE (*,10)
   10 FORMAT (//,21X,'List of constraints',/)
      DO 20 I=1,LAST-3,4
   20   WRITE (*,30) (J,ELELONG(J),J=I,I+3)
   30 FORMAT (1X,4(I2,': ',A12,2X))
      RETURN
      END
      SUBROUTINE DELECON
C
C Some or all of the constraints currently considered in the model can
C be removed.
C
$INCLUDE:'NETPATH.BLO'
   10 FORMAT (A)
      INOELE=NOELE
   20 IF (NOELE.NE.0) GO TO 40
      CALL POSCUR (-1)
```

```
      WRITE (*,30)
   30 FORMAT (' No constraints to delete.  Press <Enter> to continue.')
      READ (*,'(A)') ANS
      RETURN
   40 CALL LISTCON (INOELE)
   50 WRITE (*,60)
   60 FORMAT (//,' Enter constraint to delete. <Enter> to quit.')
      READ (*,10) BANS
      IF (BANS.EQ.' ') GO TO 70
      READ (BANS,'(i3)',ERR=50) I
      IF (I.LT.1.OR.I.GT.INOELE) GO TO 70
      IF (IELE(I).EQ.0) GO TO 20
      IELE(I)=0
      NOELE=NOELE-1
      IF (NOELE.EQ.0) GO TO 70
      GO TO 20
   70 J=0
      DO 80 I=1,INOELE
        IF (IELE(I).EQ.0) THEN
          J=J+1
        ELSE
          IELE(I-J)=IELE(I)
        END IF
   80 CONTINUE
      RETURN
      END
      SUBROUTINE DELEPHA
C
C Some or all of the phases currently considered in the model can be
C removed from consideration.
C
$INCLUDE:'NETPATH.BLO'
      IF (NOPHA.EQ.0) THEN
        CALL POSCUR (-1)
        WRITE (*,10)
   10   FORMAT (' No phases to delete.  Hit <Enter> to continue.')
        READ (*,'(a)') ANS
        RETURN
      END IF
      LEFT=NOPHA
   20 CALL LISTPHA (II)
      IF (II.GT.0.AND.II.LE.NOPHA) GO TO 50
   30 WRITE (*,40)
   40 FORMAT (//,' Input phase to delete. <Enter> to quit')
      READ (*,'(A)') BANS
      IF (BANS.EQ.' ') GO TO 60
      READ (BANS,'(I3)',ERR=30) II
      IF (II.LT.1.OR.II.GT.NOPHA) GO TO 30
   50 PHASE(II)=' '
      LEFT=LEFT-1
      IF (LEFT.GT.0) GO TO 20
   60 J=0
      DO 90 I=1,NOPHA
        IF (PHASE(I).EQ.' ') THEN
          J=J+1
        ELSE
          PHASE(I-J)=PHASE(I)
          IT(I-J)=IT(I)
          F(I-J)=F(I)
          DO 70 K=1,15
            JELE(I-J,K)=JELE(I,K)
```

```
   70       PCOEFF(I-J,K)=PCOEFF(I,K)
          DO 80 K=1,10
   80       PARA(I-J,K)=PARA(I,K)
        END IF
   90 CONTINUE
      NOPHA=LEFT
      RETURN
      END
      SUBROUTINE DELETE
C
C The specific delete routines are called from this subroutine.
C
   10 WRITE (*,20)
   20 FORMAT (' Delete (1) constraints, (2) phases, or (3) neither?')
      READ (*,*,ERR=10) I
      CALL POSCUR (-1)
      IF (I.EQ.1) CALL DELECON
      IF (I.EQ.2) CALL DELEPHA
      IF (I.EQ.1.OR.I.EQ.2) CALL SCREEN
      RETURN
      END
      SUBROUTINE DONTHAVE (I,J)
C
C The flag indicating that a particular isotopic value has been entered
C is turned off, if it is not off already.  If J=1, all the flags for
C the given phase are turned off.
C
$INCLUDE:'NETPATH.BLO'
      LOGICAL EVEN
      INTEGER PRIME(2:9)
      DATA PRIME/2,3,5,7,11,13,17,19/
      IF (J.EQ.1) THEN
        PARA(I,10)=1.
        PARA(I,1)=0.
        GO TO 20
      END IF
   10 IC=0
      IF (EVEN(PARA(I,10),PRIME(J)).EQV..TRUE.) THEN
        PARA(I,10)=PARA(I,10)/PRIME(J)
        IC=1
      END IF
      IF (IC.EQ.1) GO TO 10
   20 RETURN
      END
      LOGICAL FUNCTION DOWEHAVE(I,J)
C
C A particular flag is checked to determine whether a particular
C isotopic value has been entered.
C
$INCLUDE:'NETPATH.BLO'
      LOGICAL EVEN
      INTEGER PRIME(2:9)
      DATA PRIME/2,3,5,7,11,13,17,19/
      DOWEHAVE=EVEN(PARA(I,10),PRIME(J))
      RETURN
      END
      SUBROUTINE EDIT
C
C The various parameters of the model (everything except
C constraints and phases) can be edited.
C
```

```
$INCLUDE:'NETPATH.BLO'
   10 IEX=0
      ICO2=0
      DO 20 I=1,NOPHA
        IF (PHASE(I).EQ.'EXCHANGE') IEX=1
        IF (PHASE(I).EQ.'CO2-CH4 ') ICO2=1
   20 CONTINUE
      CALL CLS
      I=1
      WRITE (*,210) I,WFILE(1:LENS(WFILE))
      I=I+1
      WRITE (*,220) I
      I=I+1
      WRITE (*,260) I
      I=I+1
      WRITE (*,230) I,YES(IFLAG(1))
      I=I+1
      WRITE (*,240) I,WLLNMS(WELL(1))(5:LENS(WLLNMS(WELL(1))))
      IF (IFLAG(1).EQ.1) THEN
        I=I+1
        WRITE (*,240) I,WLLNMS(WELL(2))(5:LENS(WLLNMS(WELL(2))))
      END IF
      I=I+1
      WRITE (*,250) I,WLLNMS(WELL(3))(5:LENS(WLLNMS(WELL(3))))
      I=I+1
      WRITE (*,270) I,YES(IFLAG(6))
      IF (IEX.EQ.1) THEN
        I=I+1
        WRITE (*,280) I,ION(IFLAG(2))
        IF (IFLAG(2).EQ.4) THEN
          I=I+1
          WRITE (*,290) I,P(2)
        END IF
      END IF
      IF (ICO2.EQ.1) THEN
        I=I+1
        WRITE (*,300) I,P(1)
      END IF
      I=I+1
      WRITE (*,310) I
      I=I+1
      WRITE (*,320) I,YES(IFLAG(3))
      IF (IFLAG(3).EQ.1) THEN
        I=I+1
        WRITE (*,330) I
        I=I+1
        WRITE (*,340) I,MODEL(IFLAG(4))
        I=I+1
        WRITE (*,350) I,FFACT(IFLAG(5))
      END IF
   30 CALL POSCUR (I)
      CALL CLPART
      WRITE (*,360)
      READ (*,370) LINE
      IF (LINE.EQ.' ') GO TO 380
      READ (LINE,40,ERR=30) J
   40 FORMAT (I4)
      IF (IFLAG(1).EQ.0.AND.J.GT.5) J=J+1
      IF (J.GT.8.AND.IEX.EQ.0) J=J+2
      IF (J.GT.9.AND.IEX.EQ.1.AND.IFLAG(2).NE.4) J=J+1
      IF (ICO2.EQ.0.AND.J.GT.10) J=J+1
```

```
      IF (IFLAG(3).EQ.0.AND.J.GT.13) J=J+3                              AR 670
      GO TO (50,60,70,80,90,90,90,120,130,140,150,160,170,180,190,200), AR 680
     *J                                                                 AR 690
C EDIT WELL FILE                                                        AR 700
   50 IEDIT=2                                                           AR 710
      CALL WELLFILE                                                     AR 720
      GO TO 10                                                          AR 730
C EDIT MODEL                                                            AR 740
   60 IEDIT=2                                                           AR 750
      CALL MODELS                                                       AR 760
      GO TO 10                                                          AR 770
C EDIT PHASES                                                           AR 780
   70 CALL EDITPHA (I+1)                                                AR 790
      GO TO 10                                                          AR 800
C EDIT MIXING                                                           AR 810
   80 IEDIT=1                                                           AR 820
      CALL EDITMIX (I+1)                                                AR 830
      GO TO 10                                                          AR 840
C CHANGE WELLS                                                          AR 850
   90 IEDIT=1                                                           AR 860
      CALL CLS                                                          AR 870
      IF (J.LT.7) THEN                                                  AR 880
        WRITE (*,100) J-4                                               AR 890
      ELSE                                                              AR 900
        WRITE (*,110)                                                   AR 910
      END IF                                                            AR 920
      CALL WLLIST (J-4)                                                 AR 930
      GO TO 10                                                          AR 940
  100 FORMAT (21X,'Initial Well',I2,/)                                  AR 950
  110 FORMAT (23X,'Final Well',/)                                       AR 960
C EDIT EVAPORATION/DILUTION                                             AR 970
  120 CALL EDITEVAP (I+1)                                               AR 980
      GO TO 10                                                          AR 990
C EDIT ION EXCHANGE                                                     AR1000
  130 CALL EDITIONEX (I+1)                                              AR1010
      GO TO 10                                                          AR1020
C EDIT FRACTION CALCIUM IN USER-ENTERED EXCHANGE                        AR1030
  140 CALL EDITXCA (I+1)                                                AR1040
      GO TO 10                                                          AR1050
C EDIT FRACITON OF CO2 IN CO2-CH4 PHASE                                 AR1060
  150 CALL EDITXCO2 (I+1)                                               AR1070
      GO TO 10                                                          AR1080
C EDIT REDOX STATE OF DISSOLVED ORGANIC CARBON                          AR1090
  160 CALL EDITRS (I+1)                                                 AR1100
      GO TO 10                                                          AR1110
C EDIT CALCULATION OF RAYLEIGH ISOTOPIC NUMBERS                         AR1120
  170 CALL EDITCISO (I+1)                                               AR1130
      GO TO 10                                                          AR1140
C EDIT ISOTOPIC DATA                                                    AR1150
  180 CALL ISOTDATA                                                     AR1160
      GO TO 10                                                          AR1170
C EDIT C14 MODEL                                                        AR1180
  190 CALL EDITC14                                                      AR1190
      GO TO 10                                                          AR1200
C EDIT FRACTIONATION FACTORS                                            AR1210
  200 CALL EDITFACT (I+1)                                               AR1220
      GO TO 10                                                          AR1230
  210 FORMAT ('  General',/,I5,') Well file               : ',A)        AR1240
  220 FORMAT (I5,') Entire model')                                      AR1250
  230 FORMAT (' Wells',/,I5,') Mixing                  : ',A)           AR1260
  240 FORMAT (I5,') Initial well             : ',A)                     AR1270
```

```
  250 FORMAT (I5,') Final well              : ',A)                       AR1280
  260 FORMAT (I5,') Phases')                                             AR1290
  270 FORMAT ('  Parameters',/,I5,') Evaporation/Dilution    : ',A)      AR1300
  280 FORMAT (I5,') Ion exchange            : ',A)                       AR1310
  290 FORMAT (I5,') Fraction Ca             :',F6.3)                     AR1320
  300 FORMAT (I5,') Fraction CO2 in CO2-CH4 :',F6.3)                     AR1330
  310 FORMAT (I5,') Redox state of DOC')                                 AR1340
  320 FORMAT ('  Isotope calculations',/,I5,') Rayleigh calculations   : AR1350
     * ',A)                                                              AR1360
  330 FORMAT (I5,') Isotopic data')                                      AR1370
  340 FORMAT (I5,') Model for initial C14   : ',A)                       AR1380
  350 FORMAT (I5,') Carbon fract. factors   : ',A)                       AR1390
  360 FORMAT (1X,20('-'),/,' Edit which? (<Enter> when done)')           AR1400
  370 FORMAT (A)                                                         AR1410
  380 CALL SCREEN                                                        AR1420
      RETURN                                                             AR1430
      END                                                                AR1440
      SUBROUTINE EDITC14                                                 AS  10
C                                                                        AS  20
C The model to be used for the initial Carbon-14 value is selected, and  AS  30
C the parameters for it are modified.  Parameters for all the models may AS  40
C be entered because all the models may be run.                          AS  50
C                                                                        AS  60
$INCLUDE:'NETPATH.BLO'                                                   AS  70
      CHARACTER*38 C1WORDS(0:3),C2WORDS(0:3)                             AS  80
      DATA C1WORDS/'Original Value                        ','User-define AS  90
     *d Value                    ','                                     AS 100
     *','                                      '/                        AS 110
      DATA C2WORDS/'User-defined Value                    ','Mass Balanc AS 120
     *e - no fractionation       ','Mass Balance - with fractionation    AS 130
     *  ','Open System (gas-solution equilibrium)'/                      AS 140
      IDONE=0                                                            AS 150
   10 CALL CLS                                                           AS 160
      IF (IFLAG(1).EQ.0) THEN                                            AS 170
        WRITE (*,20)                                                     AS 180
      ELSE                                                               AS 190
        WRITE (*,30)                                                     AS 200
      END IF                                                             AS 210
   20 FORMAT (4X,'Initial Carbon-14, A0, (percent modern)',/,9X,'for Tot AS 220
     *al Dissolved Carbon',//,9X,'Model',15X,'Initial Well',/)           AS 230
   30 FORMAT (8X,'Initial Carbon-14, A0, (percent modern)',/,13X,'for To AS 240
     *tal Dissolved Carbon',//,9X,'Model',17X,'Init 1',4X,'Init 2',/)    AS 250
      IF (DBDATA(WELL(1),1).LE.0.0.OR.(IFLAG(1).EQ.1.AND.DBDATA(WELL(2), AS 260
     *1).LE.0.0)) THEN                                                   AS 270
        IF (DBDATA(WELL(1),1).LE.0.0) WRITE (*,40) WLLNMS(WELL(1))(5:36) AS 280
        IF (DBDATA(WELL(2),1).LE.0.0.AND.IFLAG(1).EQ.1) WRITE (*,40)     AS 290
     *   WLLNMS(WELL(2))(5:36)                                           AS 300
   40   FORMAT (' Carbon not positive for ''',A32,'''.')                 AS 310
   50   FORMAT (/,' Carbon isotopes cannot be run. Hit <Enter> to cont', AS 320
     *    'inue.')                                                       AS 330
        WRITE (*,50)                                                     AS 340
        READ (*,210) ANS                                                 AS 350
        RETURN                                                           AS 360
      END IF                                                             AS 370
      DO 60 I=1,9                                                        AS 380
        IF (IFLAG(1).EQ.0) THEN                                          AS 390
          WRITE (*,70) I,MODEL(I),C14(I,1)                               AS 400
        ELSE                                                             AS 410
          WRITE (*,70) I,MODEL(I),C14(I,1),C14(I,2)                      AS 420
        END IF                                                           AS 430
   60 CONTINUE                                                           AS 440
```

```
 70 FORMAT (I4,' : ',A,':',2(F10.2))
    IF (IFLAG(4).LT.1.OR.IFLAG(4).GT.9) IFLAG(4)=1
 80 IF (IDONE.EQ.0) WRITE (*,100) MODEL(IFLAG(4))
   *(1:LENS(MODEL(IFLAG(4))))
    IF (IDONE.EQ.1) WRITE (*,90)
 90 FORMAT (/,' Enter number of model to use (<Enter> to quit, 0 to',' 
   * edit data for all models)')
100 FORMAT (/,' Enter number of model to use (<Enter> for ''',A,''')')
    READ (*,210) ANS
    IF (ANS.EQ.' '.AND.IDONE.EQ.1) RETURN
    IF (ANS.EQ.' ') GO TO 120
    READ (ANS,110,ERR=80) I
    IF (I.EQ.0) THEN
      J=0
      CALL CLS
      GO TO 160
    END IF
    IF (I.LE.0.OR.I.GT.9) GO TO 80
    IFLAG(4)=I
110 FORMAT (I3)
120 IF (IFLAG(4).EQ.1.OR.IFLAG(4).EQ.3) THEN
      WRITE (*,130)
      READ (*,210) ANS
      IF (ANS.NE.' ') GO TO 150
      RETURN
    END IF
130 FORMAT (/,' Hit <Enter> to quit or any other key to edit data',' f
   *or all models.')
    WRITE (*,140) MODEL(IFLAG(4))(1:LENS(MODEL(IFLAG(4))))
    IDONE=1
140 FORMAT (/,' Hit <Enter> to input data for ''',A,''',',/,' any othe
   *r key to enter data for all models.')
    READ (*,210) ANS
150 CALL CLS
    J=0
    IF (ANS.EQ.' ') J=IFLAG(4)
160 IF ((J.GE.4.AND.J.LE.7).OR.J.EQ.0) THEN
      CALL INPTRL (C14DAT(1),'C-14 activity in carbonate minerals'//'
   *(% modern)')
      CALL INPTRL (C14DAT(2),'C-14 activity in soil gas CO2'//' (% mod
   *ern)')
    END IF
    IF (J.EQ.2.OR.J.EQ.0) THEN
      CALL INPTRL (C14DAT(8),'C-14 activity in dolomite (% modern)')
      CALL INPTRL (C14DAT(9),'C-14 activity in calcite (% modern)')
      IF (J.EQ.2) CALL INPTRL (C14DAT(2),'C-14 activity in soil gas CO
   *2 (% modern)')
    END IF
    IF ((J.GE.5.AND.J.LE.8).OR.J.EQ.0) THEN
      CALL INPTIN (I11,'C-13 (TDIC) in initial solution',' (Used onl
   *y in A0 models)',C1WORDS)
      IF (I11.EQ.1) CALL INPTRL (C14DAT(3),'delta C-13 (per mil) in th
   *e solution')
      CALL INPTRL (C14DAT(4),'delta C-13 (per mil) in carbonate minera
   *ls')
      CALL INPTIN (I10,'delta C-13 (per mil) in soil gas CO2',' ',
   *   C2WORDS)
      IF (I10.EQ.0) CALL INPTRL (C14DAT(5),'delta C-13 (per mil) in so
   *il gas CO2')
      IF (I10.EQ.1.OR.I10.EQ.2) THEN
        CALL INPTRL (C14DAT(12),'delta C-13 (per mil) in dolomite')
```

```
          CALL INPTRL (C14DAT(13),'delta C-13 (per mil) in calcite')
        END IF
        IF (IFLAG(1).EQ.0) THEN
          WRITE (*,170) C14(-1,1)
        ELSE
          WRITE (*,180) C14(-1,1),C14(-1,2)
        END IF
        WRITE (*,190)
        READ (*,200) LINE
      END IF
  170 FORMAT (' C-13 of CO2 gas for initial well: ',F8.3)
  180 FORMAT (' C-13 of CO2 gas for initial well 1: ',F8.3,/,' C-13 of C
     *O2 gas for initial well 2: ',F8.3)
  190 FORMAT (' Hit <Enter> to continue')
  200 FORMAT (A)
      IF (J.EQ.9) THEN
        CALL INPTRL (C14DAT(6),'user-defined C-14 activity for '//
     *   WLLNMS(WELL(1))(5:LENS(WLLNMS(WELL(1)))))
        IF (IFLAG(1).EQ.1) THEN
          CALL INPTRL (C14DAT(7),'user-defined C-14 activity for '//
     *     WLLNMS(WELL(2))(5:LENS(WLLNMS(WELL(2)))))
        END IF
      END IF
      GO TO 10
  210 FORMAT (A)
      END
      SUBROUTINE EDITCISO (IPOS)
C
C This subroutine is called as a result of Edit-Isotopes.
C Rayleigh calculations can be selected.
C
$INCLUDE:'NETPATH.BLO'
      CALL POSCUR (IPOS)
      CALL CLPART
      WRITE (*,10) YES(IFLAG(3))
   10 FORMAT (' Do Rayleigh calculations? (<Enter> for ',A3,')')
      READ (*,20) ANS
   20 FORMAT (A,A)
      IF (UPCS(ANS).EQ.'Y') IFLAG(3)=1
      IF (UPCS(ANS).EQ.'N') IFLAG(3)=0
      RETURN
      END
      SUBROUTINE EDITEVAP (IPOS)
C
C Evaporation can be considered.  This also includes dilution.
C
$INCLUDE:'NETPATH.BLO'
      CALL POSCUR (IPOS)
      CALL CLPART
      WRITE (*,10) YES(IFLAG(6))
   10 FORMAT (' Do you wish to consider evaporation?  <Enter> for ',A)
      I=0
      IF (YN(YES(IFLAG(6)))) I=1
      IFLAG(6)=I
      RETURN
      END
      SUBROUTINE EDITFACT (IPOS)
$INCLUDE:'NETPATH.BLO'
   10 CALL POSCUR (IPOS)
      CALL CLPART
```

```
      WRITE (*,20) (FFACT(J),J=0,1),FFACT(IFLAG(5))
   20 FORMAT (' Enter fractionation factors to use: 1) ',A4,' or 2) ',
     *A14,/,' <Enter> for ',A)
      READ (*,30) ANS
   30 FORMAT (A)
      IF (ANS.EQ.' ') RETURN
      READ (ANS,40,ERR=10) J
   40 FORMAT (I1)
      IF (J.LT.1.OR.J.GT.2) GO TO 10
      IFLAG(5)=J-1
      RETURN
      END
      SUBROUTINE EDITIONEX (IPOS)
C
C The specific exchange to be used under the general EXCHANGE phase is
C selected.  If Variable Ca/Mg is selected, the fraction of Ca in the
C exchange is entered.
C
$INCLUDE:'NETPATH.BLO'
   10 CALL POSCUR (IPOS)
      CALL CLPART
      WRITE (*,20) (ION(I),I=1,4),ION(IFLAG(2))
   20 FORMAT (' Select exchange: (1) ',A8,', (2) ',A5,', (3) ',A5,', or
     *(4) ',A10,/,' <Enter> for ',A10)
      READ (*,30) LINE
   30 FORMAT (A,A)
      IF (LINE.EQ.' ') GO TO 40
      READ (LINE,'(i2)',ERR=10) I2
      IFLAG(2)=I2
   40 IF (IFLAG(2).EQ.4) CALL EDITXCA (IPOS)
      RETURN
      END
      SUBROUTINE EDITMIX (IPOS)
$INCLUDE:'NETPATH.BLO'
      CALL POSCUR (IPOS)
      CALL CLPART
      IF (IEDIT.EQ.0) THEN
        WRITE (*,10)
   10   FORMAT (' Is this a mixing problem?')
        IF (YN('N')) IFLAG(1)=1
      ELSE
        WRITE (*,20) YES(IFLAG(1))
   20   FORMAT (' Is this a mixing problem? <Enter> for ',A3)
        IFL=IFLAG(1)
        IFLAG(1)=0
        IF (YN(YES(IFL)(1:1))) IFLAG(1)=1
        IF (IFL.EQ.0.AND.IFLAG(1).EQ.1) THEN
          CALL CLS
          WRITE (*,30) 2
   30     FORMAT (21X,'Initial Well',I2,/)
          IEDIT=0
          CALL WLLIST (2)
        END IF
      END IF
      RETURN
      END
      SUBROUTINE EDITPARA
C
C The parameter to be edited is selected.
C
      CALL POSCUR (-1)
```

```
      WRITE (*,10)
   10 FORMAT (' Edit (1) Ion exchange, (2) XCO2, (3) Isotopes,',' (4) Ev
     *aporation, (5) RS of DOC,',/,'   or (6) abort')
      READ (*,*,ERR=20) I
      IF (I.EQ.1) CALL EDITIONEX (-1)
      IF (I.EQ.2) CALL EDITXCO2 (-1)
      IF (I.EQ.3) CALL EDITCISO (-1)
      IF (I.EQ.4) CALL EDITEVAP (-1)
      IF (I.EQ.5) CALL EDITRS (-1)
      IF (I.GE.1.AND.I.LE.5) CALL SCREEN
   20 RETURN
      END
      SUBROUTINE EDITPHA (IPOS)
C
C The phase to be edited is selected and ADDPHA is called.  The actual
C editing occurs in either ADDPHA or EDITPIEC.
C
$INCLUDE:'NETPATH.BLO'
      IF (NOPHA.EQ.0) THEN
        CALL POSCUR (IPOS)
        WRITE (*,10)
   10   FORMAT (' No phases to edit.  Hit <Enter> to continue.')
        READ (*,'(a3)') BANS
        RETURN
      END IF
      CALL LISTPHA (IJ)
      IF (IJ.EQ.0) THEN
   20   WRITE (*,30)
   30   FORMAT (/,' Input phase to edit.  <Enter to abort>')
        READ (*,'(A3)') BANS
        IF (BANS.EQ.' ') RETURN
        READ (BANS,'(I3)',ERR=20) IJ
        IF (IJ.LE.0.OR.IJ.GT.NOPHA) GO TO 20
      END IF
      IEDIT=1
   40 CALL CLS
      IF (FLIN.EQ.1) THEN
        CALL EDITPIEC (IJ)
        RETURN
      END IF
      WRITE (*,50) PHASE(IJ)
   50 FORMAT (' Phase editing choices for ',A,/,6X,'1) Replace phase ent
     *irely',/,6X,'2) Edit phase composition',/,6X,'3) Edit forcing and
     *dissolution/precipitation only',/,' Enter your choice, or <Enter>
     *to abort the edit:')
      READ (*,60) IE
   60 FORMAT (I1)
      IF (IE.LT.1.OR.IE.GT.3) GO TO 40
      IF (IE.EQ.2) THEN
        CALL EDITPIEC (IJ)
        RETURN
      END IF
      IF (IE.EQ.3) THEN
        CALL TRANS (IJ)
        RETURN
      END IF
      CALL ADDPHA (IJ)
      RETURN
      END
```

```
      SUBROUTINE EDITPIEC (IJ)
C
C A new phase is entered from scratch or the name and constraints of a
C current phase may be edited.
C
$INCLUDE:'NETPATH.BLO'
      CHARACTER*8 MIDDLE
      IF (IEDIT.EQ.0) GO TO 30
      WRITE (*,10) PHASE(IJ)
   10 FORMAT (/,' Enter phase name. <Enter> for ''',A,'''.')
      READ (*,20) MIDDLE
   20 FORMAT (A)
      IF (MIDDLE.NE.' ') PHASE(IJ)=MIDDLE
      GO TO 50
   30 WRITE (*,40)
   40 FORMAT (/,' Enter phase name. <Enter> to abort.')
      READ (*,20) MIDDLE
      IF (MIDDLE.EQ.' ') RETURN
      PHASE(IJ)=MIDDLE
      CALL DONTHAVE (IJ,1)
      JELE(IJ,1)=0
   50 I=0
   60 I=I+1
      IF (JELE(IJ,I).NE.0) GO TO 60
      NUM=I-1
      CALL CLS
      WRITE (*,70) PHASE(IJ)
   70 FORMAT ('    Data for phase ''',A,'''.',/)
      DO 80 I=1,NUM
   80   WRITE (*,90) I,ELELONG(JELE(IJ,I)),PCOEFF(IJ,I)
   90 FORMAT (I3,': ',A12,'............',F8.3)
      IF (NUM.EQ.0) GO TO 110
      WRITE (*,100)
  100 FORMAT (/,' <A>dd, <D>elete, or <E>dit a constraint, <Enter> to','
     * continue.')
      READ (*,20) BANS
      BANS=UPCS(BANS)
      IF (BANS.EQ.' ') GO TO 320
      IF (BANS.EQ.'A'.AND.I.EQ.14) GO TO 50
      IF (BANS.EQ.'A') GO TO 110
      IF (BANS.EQ.'D') GO TO 200
      IF (BANS.EQ.'E') GO TO 230
      GO TO 50
  110 CALL CONLIST (20)
  120 WRITE (*,130)
  130 FORMAT (/,' Input number of constraint to add. <Enter> to stop,','
     *'''?'' for the list.')
      READ (*,20) BANS
      IF (BANS.EQ.'?') GO TO 110
      IF (BANS.EQ.' ') GO TO 180
      READ (BANS,'(i3)',ERR=120) II
      IF (II.LT.1.OR.II.GT.20) GO TO 110
      I=0
  140 I=I+1
      IF (JELE(IJ,I).EQ.II) GO TO 160
      IF (JELE(IJ,I).NE.0) GO TO 140
      NUM=NUM+1
      JELE(IJ,NUM)=II
      JELE(IJ,NUM+1)=0
      WRITE (*,150) ELELONG(II)
  150 FORMAT (/,' Input coefficient of ''',A,'''.')
```

```
      READ (*,*) COEF                                                   BA 620
      PCOEFF(IJ,I)=COEF                                                 BA 630
      GO TO 120                                                         BA 640
  160 WRITE (*,170)                                                     BA 650
  170 FORMAT (/,' ERROR - Already a coefficient.')                      BA 660
      GO TO 120                                                         BA 670
  180 I=0                                                               BA 680
  190 I=I+1                                                             BA 690
      IF (JELE(IJ,I).NE.0) GO TO 190                                    BA 700
      NUM=I-1                                                           BA 710
      GO TO 50                                                          BA 720
  200 WRITE (*,210)                                                     BA 730
  210 FORMAT (/,' Input number of constraint to delete. (','<Enter> to qBA 740
     *uit)')                                                            BA 750
      READ (*,20) BANS                                                  BA 760
      IF (BANS.EQ.' ') GO TO 50                                         BA 770
      READ (BANS,'(i3)',ERR=200) II                                     BA 780
      IF (II.LT.1.OR.II.GT.NUM) GO TO 200                               BA 790
      DO 220 I=II,NUM-1                                                 BA 800
        JELE(IJ,I)=JELE(IJ,I+1)                                         BA 810
  220   PCOEFF(IJ,I)=PCOEFF(IJ,I+1)                                     BA 820
      JELE(IJ,NUM)=0                                                    BA 830
      NUM=NUM-1                                                         BA 840
      GO TO 50                                                          BA 850
  230 WRITE (*,240)                                                     BA 860
  240 FORMAT (/,' Which coefficient to edit? <Enter> to quit')          BA 870
      READ (*,20) BANS                                                  BA 880
      IF (BANS.EQ.' ') GO TO 50                                         BA 890
      READ (BANS,'(i3)',ERR=230) II                                     BA 900
      IF (II.LT.1.OR.II.GT.NUM) GO TO 230                               BA 910
      CALL CONLIST (20)                                                 BA 920
  250 WRITE (*,260) ELELONG(JELE(IJ,II))                                BA 930
  260 FORMAT (/,' Which constraint to include? <Enter> keeps ''',A,'''.'BA 940
     *)                                                                 BA 950
      READ (*,20) BANS                                                  BA 960
      IF (BANS.EQ.' ') GO TO 280                                        BA 970
      READ (BANS,'(i3)',ERR=250) I                                      BA 980
      IF (I.LT.1.OR.I.GT.20) GO TO 250                                  BA 990
      J=0                                                               BA1000
  270 J=J+1                                                             BA1010
      IF (J.EQ.II) GO TO 270                                            BA1020
      IF (JELE(IJ,J).EQ.I) GO TO 300                                    BA1030
      IF (JELE(IJ,J).NE.0) GO TO 270                                    BA1040
      JELE(IJ,II)=I                                                     BA1050
  280 WRITE (*,290) ELELONG(JELE(IJ,II)),PCOEFF(IJ,II)                  BA1060
  290 FORMAT (' Input coefficient for ',A,/,' <Enter> for ',F7.2)       BA1070
      READ (*,20) MIDDLE                                                BA1080
      IF (MIDDLE.EQ.' ') GO TO 50                                       BA1090
      READ (MIDDLE,'(f8.0)',ERR=280) COEF                               BA1100
      PCOEFF(IJ,II)=COEF                                                BA1110
      GO TO 50                                                          BA1120
  300 WRITE (*,310)                                                     BA1130
  310 FORMAT (' ERROR - Already included in this phase.')               BA1140
      GO TO 250                                                         BA1150
  320 IF (IEDIT.EQ.0) THEN                                              BA1160
        IT(IJ)=' '                                                      BA1170
        F(IJ)=' '                                                       BA1180
      END IF                                                            BA1190
      CALL TRANS (IJ)                                                   BA1200
      CALL SAVEOTHER (IJ)                                               BA1210
      RETURN                                                            BA1220
```

```
      END
      SUBROUTINE EDITRS (IPOS)
$INCLUDE:'NETPATH.BLO'
      DO 30 J=1,3
        IF (IFLAG(1).EQ.0.AND.J.EQ.2) GO TO 30
   10   CALL POSCUR (IPOS)
        CALL CLPART
        IF (NODATA(WELL(J),48).EQ.0) THEN
          IF (ISDOCRS(J).EQ.0) THEN
            WRITE (*,40) WLLNMS(WELL(J))(5:LENS(WLLNMS(WELL(J)))),
     *       DBDATA(WELL(J),48)
          ELSE
            WRITE (*,50) WLLNMS(WELL(J))(5:LENS(WLLNMS(WELL(J)))),
     *       DBDATA(WELL(J),48),DBDATA(WELL(J),49)
          END IF
        ELSE
          IF (ISDOCRS(J).EQ.0) THEN
            WRITE (*,60) WLLNMS(WELL(J))(5:LENS(WLLNMS(WELL(J))))
          ELSE
            WRITE (*,70) WLLNMS(WELL(J))(5:LENS(WLLNMS(WELL(J)))),
     *       DBDATA(WELL(J),49)
          END IF
        END IF
        READ (*,80) ANS
        ANS=UPCS(ANS)
        IF (ANS.NE.' '.AND.ANS.NE.'Y'.AND.ANS.NE.'N') GO TO 10
        IF (ANS.EQ.'N') THEN
          ISDOCRS(J)=0
          GO TO 30
        END IF
        IF (ANS.EQ.' '.AND.ISDOCRS(J).EQ.0) GO TO 30
   20   CALL POSCUR (IPOS)
        CALL CLPART
        IF (ISDOCRS(J).EQ.0) THEN
          WRITE (*,90) WLLNMS(WELL(J))(5:)
        ELSE
          WRITE (*,100) WLLNMS(WELL(J))(5:),DBDATA(WELL(J),49)
        END IF
        READ (*,110) LINE
        IF (LINE.EQ.' ') THEN
          IF (ISDOCRS(J).EQ.0) THEN
            GO TO 20
          ELSE
            GO TO 30
          END IF
        END IF
        READ (LINE,120,ERR=20) ADOCRS
        DBDATA(WELL(J),49)=ADOCRS
        ISDOCRS(J)=1
   30 CONTINUE
   40 FORMAT (' The orginal value for RS of DOC in ',A,/,'  is ',F8.2,'.
     *',' Do you want to enter a new value? (<Enter> = no)')
   50 FORMAT (' The orginal value for RS of DOC in ',A,/,'  is ',F8.2,'.
     *',' The user-entered value is ',F8.2,'.',/,' Do you want to use a
     *user-entered value? (<Enter> = yes)')
   60 FORMAT (' The orginal value for RS of DOC in ',A,/,'  is undefined
     *. Do you want to enter a new value?',' (<Enter> = no)')
   70 FORMAT (' The orginal value for RS of DOC in ',A,/,'  is undefined
     *. The user-entered value is ',F8.2,'.',/,' Do you want to use a us
     *er-entered value? (<Enter> = yes)')
```

```
   80 FORMAT (A)
   90 FORMAT (' Enter value for redox state of DOC for ',A)
  100 FORMAT (' Enter value for redox state of DOC for ',A,/,' <Enter> f
     *or ',F8.2)
  110 FORMAT (A80)
  120 FORMAT (F10.0)
      RETURN
      END
      SUBROUTINE EDITXCA (IPOS)
$INCLUDE:'NETPATH.BLO'
   10 CALL POSCUR (IPOS)
      CALL CLPART
      IF (IEDIT.EQ.0) THEN
        WRITE (*,20)
      ELSE
        WRITE (*,20) P(2)
      END IF
   20 FORMAT (' Enter fraction Ca in Ca/Mg.',:,'  <Enter> for ',F6.3)
      READ (*,30) LINE
      IF (LINE.EQ.' '.AND.IEDIT.EQ.0) GO TO 10
      IF (LINE.EQ.' ') RETURN
      READ (LINE,'(f20.0)',ERR=10) P2
      IF (P2.LT.0.0.OR.P2.GT.1.0) GO TO 10
      P(2)=P2
   30 FORMAT (A)
      RETURN
      END
      SUBROUTINE EDITXCO2 (IPOS)
C
C CO2 gas and CH4 gas can be considered as one phase in NETPATH.  Here,
C the fraction of CO2 in the mixture is entered.
C
$INCLUDE:'NETPATH.BLO'
   10 CALL POSCUR (IPOS)
      CALL CLPART
      WRITE (*,20) P(1)
   20 FORMAT (' Enter fraction CO2 in CO2-CH4.  <Enter> for ',F6.3)
      READ (*,30) LINE
   30 FORMAT (A,A)
      IF (LINE.EQ.' ') RETURN
      READ (LINE,'(f20.0)',ERR=10) P1
      IF (P1.LT.0.0.OR.P1.GT.1.0) GO TO 10
      P(1)=P1
      RETURN
      END
      LOGICAL FUNCTION EVEN(ADS,I)
C
C ADS is checked for divisibility by I.  Used in system to record which
C isotopic values have been entered.
C
      DOUBLE PRECISION ADS
      J=ADS
      EVEN=.FALSE.
      IF (MOD(J,I).EQ.0) EVEN=.TRUE.
      RETURN
      END
      FUNCTION FINDTOT (I)
C
C This function is needed because a reordering of wells in the well file
C would cause incorrect wells to be used for a model unless permanent
C numbers of wells are stored.  Here, the well that corresponds to a
```

```
C particular permanent number is located.
C
$INCLUDE:'NETPATH.BLO'
      J=0
      FINDTOT=0
   10 J=J+1
      IF (TOT(J).EQ.I) THEN
        FINDTOT=J
        RETURN
      END IF
      IF (J.LT.NWLLS) GO TO 10
      RETURN
      END
      SUBROUTINE HAVE (I,J)
C
C The flag indicating that a particular isotopic value has been entered
C is set, if it has not been set already.
C
$INCLUDE:'NETPATH.BLO'
      LOGICAL EVEN
      INTEGER PRIME(2:9)
      DATA PRIME/2,3,5,7,11,13,17,19/
      IF (EVEN(PARA(I,10),PRIME(J)).EQV..FALSE.) PARA(I,10)=PARA(I,10)*
     *PRIME(J)
      RETURN
      END
      SUBROUTINE ICCARBON
C
C The isotopic compositions of methane and DOC are edited for each of
C the currently chosen wells, but only if DOC and methane numbers were
C entered.
C
$INCLUDE:'NETPATH.BLO'
      INTEGER ICHAVE(3,2),ITHIS(3)
      JHAVE=0
      DO 20 I=1,3
        DO 10 J=1,2
          ICHAVE(I,J)=0
          IF (I.EQ.2.AND.IFLAG(1).EQ.0) GO TO 10
          IF (DBDATA(WELL(I),41+J).GT.0) ICHAVE(I,J)=1
          IF (ICHAVE(I,J).EQ.1) JHAVE=1
   10   CONTINUE
   20 CONTINUE
      IF (JHAVE.EQ.0) RETURN
   30 CALL CLS
      WRITE (*,40)
   40 FORMAT (19X,'Isotopic compositions of Carbon in solution',/)
      WRITE (*,50)
   50 FORMAT (36X,' Carbon-13  C14 %mod   Carbon-13  C14 %mod',/,' #   ',
     *'Well Name',23X,'  of CH4     of CH4     of DOC','       of DOC',/,
     *1X,78('-'))
      J=0
      DO 60 I=1,3
        IF (ICHAVE(I,1).EQ.0.AND.ICHAVE(I,2).EQ.0) GO TO 60
        J=J+1
        ITHIS(J)=I
        WRITE (LINE,140) J,WLLNMS(WELL(I))(5:36)
        IF (ICHAVE(I,1).EQ.1) WRITE (LINE,150) LINE(1:35),DBDATA(WELL(I)
     *   ,44),DBDATA(WELL(I),46)
        IF (ICHAVE(I,2).EQ.1) WRITE (LINE,150) LINE(1:57),DBDATA(WELL(I)
     *   ,45),DBDATA(WELL(I),47)
```

```
        WRITE (*,190) LINE(1:LENS(LINE))
   60 CONTINUE
      WRITE (*,70)
   70 FORMAT (/,' Enter number of well to change, <Enter>',' when done.'
     *)
      READ (*,180) LINE
      IF (LINE.EQ.' ') RETURN
      READ (LINE,80,ERR=30) I
   80 FORMAT (I10)
      IF (I.LE.0.OR.I.GT.J) GO TO 30
      I=ITHIS(I)
      IF (ICHAVE(I,1).EQ.1) THEN
   90   CALL POSCUR (J+2)
        CALL CLPART
        WRITE (*,160) 'Carbon-13','CH4',DBDATA(WELL(I),44)
        READ (*,180) LINE
        IF (LINE.EQ.' ') GO TO 100
        READ (LINE,170,ERR=90) AIN
        DBDATA(WELL(I),44)=AIN
  100   CALL POSCUR (J+2)
        CALL CLPART
        WRITE (*,160) 'Carbon-14','CH4',DBDATA(WELL(I),46)
        READ (*,180) LINE
        IF (LINE.EQ.' ') GO TO 110
        READ (LINE,170,ERR=100) AIN
        DBDATA(WELL(I),46)=AIN
      END IF
  110 IF (ICHAVE(I,2).EQ.1) THEN
  120   CALL POSCUR (J+2)
        CALL CLPART
        WRITE (*,160) 'Carbon-13','DOC',DBDATA(WELL(I),45)
        READ (*,180) LINE
        IF (LINE.EQ.' ') GO TO 130
        READ (LINE,170,ERR=120) AIN
        DBDATA(WELL(I),45)=AIN
  130   CALL POSCUR (J+2)
        CALL CLPART
        WRITE (*,160) 'Carbon-14','DOC',DBDATA(WELL(I),47)
        READ (*,180) LINE
        IF (LINE.EQ.' ') GO TO 30
        READ (LINE,170,ERR=130) AIN
        DBDATA(WELL(I),47)=AIN
      END IF
      GO TO 30
  140 FORMAT (I1,': ',A32)
  150 FORMAT (A,2F10.3)
  160 FORMAT (' Enter ',A,' value for ',A,', <Enter> for ',F10.3,'.')
  170 FORMAT (F20.0)
  180 FORMAT (A)
  190 FORMAT (1X,A)
      END
      SUBROUTINE INCLISO (JE,PC,I,J,IRUN)
C
C The isotopic values (isotopic composition and fractionation factors,
C where applicable) are included with a phase.  This is used to store
C the data with a phase when saving a model.  It is also used to include
C isotope data with the phases when the model is run.
C
$INCLUDE:'NETPATH.BLO'
      LOGICAL DOWEHAVE
      INTEGER JE(15)
```

```
DOUBLE PRECISION PC(15)                                           BI 110
IEND=J                                                            BI 120
IF (IRUN.EQ.1.AND.PHASE(I).EQ.'CO2-CH4 ') GO TO 20                BI 130
DO 10 K=1,J                                                       BI 140
  IF (JE(K).EQ.1) THEN                                            BI 150
    Q=PC(K)                                                       BI 160
    IF (IRUN.EQ.0) Q=1                                            BI 170
    IF (DOWEHAVE(I,2).EQV..TRUE.) THEN                            BI 180
      JE(IEND+1)=21                                               BI 190
      PC(IEND+1)=PARA(I,2)*Q                                      BI 200
      IEND=IEND+1                                                 BI 210
      JE(IEND+1)=0                                                BI 220
    END IF                                                        BI 230
    IF (DOWEHAVE(I,3).EQV..TRUE.) THEN                            BI 240
      JE(IEND+1)=22                                               BI 250
      PC(IEND+1)=PARA(I,3)*Q                                      BI 260
      IEND=IEND+1                                                 BI 270
      JE(IEND+1)=0                                                BI 280
    END IF                                                        BI 290
    IF (DOWEHAVE(I,4).EQV..TRUE.) THEN                            BI 300
      JE(IEND+1)=23                                               BI 310
      PC(IEND+1)=PARA(I,4)*Q                                      BI 320
      IEND=IEND+1                                                 BI 330
      JE(IEND+1)=0                                                BI 340
    END IF                                                        BI 350
    IF (DOWEHAVE(I,5).EQV..TRUE.) THEN                            BI 360
      JE(IEND+1)=24                                               BI 370
      PC(IEND+1)=PARA(I,5)*Q                                      BI 380
      IEND=IEND+1                                                 BI 390
      JE(IEND+1)=0                                                BI 400
    END IF                                                        BI 410
    IF ((DOWEHAVE(I,6).EQV..TRUE.).AND.IRUN.EQ.0) THEN            BI 420
      JE(IEND+1)=29                                               BI 430
      PC(IEND+1)=PARA(I,6)                                        BI 440
      IEND=IEND+1                                                 BI 450
      JE(IEND+1)=0                                                BI 460
    END IF                                                        BI 470
    IF ((DOWEHAVE(I,7).EQV..TRUE.).AND.IRUN.EQ.0) THEN            BI 480
      JE(IEND+1)=30                                               BI 490
      PC(IEND+1)=PARA(I,7)                                        BI 500
      IEND=IEND+1                                                 BI 510
      JE(IEND+1)=0                                                BI 520
    END IF                                                        BI 530
    IF ((DOWEHAVE(I,8).EQV..TRUE.).AND.IRUN.EQ.0) THEN            BI 540
      JE(IEND+1)=31                                               BI 550
      PC(IEND+1)=PARA(I,8)                                        BI 560
      IEND=IEND+1                                                 BI 570
      JE(IEND+1)=0                                                BI 580
    END IF                                                        BI 590
    IF ((DOWEHAVE(I,9).EQV..TRUE.).AND.IRUN.EQ.0) THEN            BI 600
      JE(IEND+1)=32                                               BI 610
      PC(IEND+1)=PARA(I,9)                                        BI 620
      IEND=IEND+1                                                 BI 630
      JE(IEND+1)=0                                                BI 640
    END IF                                                        BI 650
  END IF                                                          BI 660
  IF (JE(K).EQ.2) THEN                                            BI 670
    Q=PC(K)                                                       BI 680
    IF (IRUN.EQ.0) Q=1                                            BI 690
    IF (DOWEHAVE(I,4).EQV..TRUE.) THEN                            BI 700
      JE(IEND+1)=23                                               BI 710
```

```
            PC(IEND+1)=PARA(I,4)*Q                                      BI 720
            IEND=IEND+1                                                 BI 730
            JE(IEND+1)=0                                                BI 740
          END IF                                                        BI 750
          IF ((DOWEHAVE(I,8).EQV..TRUE.).AND.IRUN.EQ.0) THEN            BI 760
            JE(IEND+1)=31                                               BI 770
            PC(IEND+1)=PARA(I,8)                                        BI 780
            IEND=IEND+1                                                 BI 790
            JE(IEND+1)=0                                                BI 800
          END IF                                                        BI 810
        END IF                                                          BI 820
        IF (JE(K).EQ.15) THEN                                           BI 830
          Q=PC(K)                                                       BI 840
          IF (IRUN.EQ.0) Q=1                                            BI 850
          IF (DOWEHAVE(I,5).EQV..TRUE.) THEN                            BI 860
            JE(IEND+1)=24                                               BI 870
            PC(IEND+1)=PARA(I,5)*Q                                      BI 880
            IEND=IEND+1                                                 BI 890
            JE(IEND+1)=0                                                BI 900
          END IF                                                        BI 910
          IF ((DOWEHAVE(I,9).EQV..TRUE.).AND.IRUN.EQ.0) THEN            BI 920
            JE(IEND+1)=32                                               BI 930
            PC(IEND+1)=PARA(I,9)                                        BI 940
            IEND=IEND+1                                                 BI 950
            JE(IEND+1)=0                                                BI 960
          END IF                                                        BI 970
        END IF                                                          BI 980
   10 CONTINUE                                                          BI 990
      J=IEND                                                            BI1000
      RETURN                                                            BI1010
   20 PC(3)=PARA(I,2)*P(1)+PARA(I,4)*(1.-P(1))                          BI1020
      PC(4)=PARA(I,3)*P(1)+PARA(I,5)*(1.-P(1))                          BI1030
      JE(3)=21                                                          BI1040
      JE(4)=22                                                          BI1050
      JE(5)=0                                                           BI1060
      J=4                                                               BI1070
      RETURN                                                            BI1080
      END                                                               BI1090
      SUBROUTINE INIT                                                   BJ  10
$INCLUDE:'NETPATH.BLO'                                                  BJ  20
C                                                                       BJ  30
C             SET PHASES 1 AND 2 IN MIXING CASE                         BJ  40
C             ALSO, HANDLE EVAPORATION AS IF WATER MIXING (IN OR OUT)   BJ  50
C                                                                       BJ  60
      IF (IMIX.EQ.1.OR.IFLAG(6).EQ.1) THEN                              BJ  70
        K=1+IMIX+IFLAG(6)                                               BJ  80
        PNAME(1)='INIT 1'                                               BJ  90
        IF (IMIX.EQ.1) PNAME(2)='INIT 2'                                BJ 100
        IF (IFLAG(6).EQ.1) PNAME(K)='WATER'                             BJ 110
        ENAME(NEQ)='MIX'                                                BJ 120
        EDELTA(NEQ)=1.0                                                 BJ 130
        PELT(1,NEQ)='MIX'                                               BJ 140
        PCOEF(1,NEQ)=1.0                                                BJ 150
        PELT(2,NEQ)='MIX'                                               BJ 160
        PCOEF(2,NEQ)=1.0                                                BJ 170
        IF (K.GT.2) THEN                                                BJ 180
          PELT(3,NEQ)='MIX'                                             BJ 190
          PCOEF(3,NEQ)=1.0                                              BJ 200
        END IF                                                          BJ 210
        DO 10 I=1,NELTS                                                 BJ 220
          PELT(1,I)=ENAME(I)                                            BJ 230
```

```
          PCOEF(1,I)=SINIT1(I)
          IF (IMIX.EQ.1) THEN
            PELT(2,I)=ENAME(I)
            PCOEF(2,I)=SINIT2(I)
          END IF
          IF (IFLAG(6).EQ.1) THEN
            PELT(K,I)=ENAME(I)
            PCOEF(K,I)=0.0
          END IF
   10   CONTINUE
      END IF
C
C             CALCALUTE ROW NUMBERS OF ELEMENTS
C
      DO 40 I=1,NMINS
        DO 30 L=1,39
          LELT(I,L)=0
          DO 20 K=1,NEQ
            IF (PELT(I,L).EQ.ENAME(K)) LELT(I,L)=K
   20     CONTINUE
   30   CONTINUE
   40 CONTINUE
      IF (IMIX.EQ.1) THEN
        TRANSFER(1)='+'
        TRANSFER(2)='+'
        FORCE(1)='F'
        FORCE(2)='F'
      END IF
      IF (IMIX.EQ.0.AND.IFLAG(6).EQ.1) THEN
        TRANSFER(1)='+'
        TRANSFER(2)=' '
        FORCE(1)='F'
        FORCE(2)='F'
      END IF
      IF (IMIX.EQ.1.AND.IFLAG(6).EQ.1) THEN
        FORCE(3)='F'
        TRANSFER(3)=' '
      END IF
      DO 50 I=1,NMINS
        IF (TRANSFER(I).EQ.'+'.OR.TRANSFER(I).EQ.'D'.OR.TRANSFER(I).EQ.'
     *d') THEN
          TRANSFER(I)='+'
        ELSE IF (TRANSFER(I).EQ.'-'.OR.TRANSFER(I).EQ.'P'.OR.TRANSFER(I)
     *   .EQ.'p') THEN
          TRANSFER(I)='-'
        ELSE IF (TRANSFER(I).EQ.' ') THEN
          TRANSFER(I)=' '
        ELSE
        END IF
   50 CONTINUE
      RETURN
      END
      SUBROUTINE INITVALS (INEW)
C
C Variables that need initial values are initialized (and probably some
C that don't).  Also, the data from NETPATH.DAT, if any, are read in.
C
$INCLUDE:'NETPATH.BLO'
      IF (INEW.EQ.1) GO TO 60
      OPEN (7,FILE='NETPATH.DAT',STATUS='OLD',ERR=110)
      I=1
```

```
   10 READ (7,20,END=40) FLINE(I)
   20 FORMAT (A,A)
      IF (FLINE(I)(1:10).NE.'          ') GO TO 30
      I=I-1
      FLINE(I)=FLINE(I)(1:80)//FLINE(I+1)(11:80)
   30 I=I+1
      IF (I.LE.100) GO TO 10
   40 FLIN=I-1
   50 IF (FLIN.EQ.0) THEN
        FLINE(1)='*OTHER**'
        FLIN=1
      END IF
      IF (FLINE(FLIN)(1:8).NE.'*OTHER**') THEN
        FLIN=FLIN+1
        FLINE(FLIN)='*OTHER**'
      END IF
      CLOSE (7)
      ESC=CHAR(27)
      WFILE=' '
   60 IEDIT=0
      NOELE=0
      NOPHA=0
      IFLAG(1)=0
      IFLAG(2)=3
      IFLAG(3)=0
      IFLAG(4)=1
      IFLAG(5)=0
      IFLAG(6)=0
      ISDOCRS(1)=0
      ISDOCRS(2)=0
      ISDOCRS(3)=0
      EVAP=0.0
      DO 100 III=1,39
        DO 70 KKK=1,10
          PARA(III,KKK)=0.
   70   CONTINUE
        DO 80 LLL=1,15
          JELE(III,LLL)=0
   80   CONTINUE
        DO 90 JJJ=1,39
          PCOEF(III,JJJ)=0.
          PELT(III,JJJ)=' '
   90   CONTINUE
  100 CONTINUE
      EFILE=' '
      PFILE=' '
      P(1)=1.0
      P(2)=0.
      DISALONG=1.
      WELL(1)=0
      WELL(2)=0
      WELL(3)=0
      WLLNMS(0)='    *UNDEFINED*'
      RETURN
  110 FLIN=0
      WRITE (*,120)
  120 FORMAT (' Warning - There is no NETPATH.DAT file.'/' An empty file
     *',' will be created. To use an existing mineral file, please',' qu
     *it'/' and copy the NETPATH.DAT file into your directory.'/,' Do yo
     *u wish to continue (<Enter>=yes)')
      IF (YN('Y').EQV..TRUE.) GO TO 50
```

```
      END
      SUBROUTINE INPTIN (CNUM,PROMPT,PROMPT2,CHOICES)
C
C Allows selection of one of a group of choices.  Called by EDITC14.
C
      INTEGER CNUM
      CHARACTER*38 CHOICES(0:3)
      CHARACTER*(*) PROMPT,PROMPT2
      CHARACTER ANS
   10 WRITE (*,20) PROMPT
   20 FORMAT (' Choices for ',A)
      IF (LENS(PROMPT2).GT.0) WRITE (*,30) PROMPT2
   30 FORMAT (A)
      MAX=-1
   40 MAX=MAX+1
      IF (MAX.LE.3) THEN
        IF (LENS(CHOICES(MAX)).GT.0) GO TO 40
      END IF
      MAX=MAX-1
      DO 50 J=0,MAX
   50   WRITE (*,60) J,CHOICES(J)
      WRITE (*,70) PROMPT,CHOICES(CNUM)(1:LENS(CHOICES(CNUM)))
      READ (*,80) ANS
      IF (ANS.EQ.' ') RETURN
      READ (ANS,'(I1)',ERR=10) I
      IF (I.LT.0.OR.I.GT.MAX) GO TO 10
      CNUM=I
   60 FORMAT (I3,' : ',A)
   70 FORMAT (/' Choose method for defining ',A,',',/,' <Enter> to use '
     *,A,'.')
   80 FORMAT (A1)
      RETURN
      END
      SUBROUTINE INPTRL (CNUM,PROMPT)
C
C A real value is input following the display of the proper prompt.
C Called by EDITC14.
C
      CHARACTER*(*) PROMPT
      CHARACTER*80 ANS
      DOUBLE PRECISION CNUM,C
   10 WRITE (*,20) PROMPT,CNUM
      READ (*,30) ANS
      IF (ANS.EQ.' ') RETURN
      READ (ANS,'(F20.0)',ERR=10) C
      CNUM=C
   20 FORMAT (/' Enter value of ',A,',',/,' <Enter> for',F10.2)
   30 FORMAT (A80)
      RETURN
      END
      SUBROUTINE ISOTDATA
C
C This subroutine is called as a result of Edit-Isotope Data
C Isotopic compositions and fractionation factors of applicable phases
C may be entered. In addition, C-13 and C-14 data for CH4 and DOC in the
C wells is also entered here.
C
$INCLUDE:'NETPATH.BLO'
      LOGICAL DOWEHAVE
      INTEGER ICCC(39)
      CHARACTER*15 C4MOD(4)
```

```
      CHARACTER*43 TITLE(0:1)                                            BN 120
      CHARACTER*29 TITLE2(0:1)                                           BN 130
      CHARACTER*9 LABEL(4)                                               BN 140
      CHARACTER*3 NAME(4)                                                BN 150
      CHARACTER*7 SULF(2)                                                BN 160
      CHARACTER*80 LINE2                                                 BN 170
      DATA C4MOD/'Original data  ','Carbonate model','Silicate model ',' BN 180
     *User defined   '/                                                  BN 190
      DATA TITLE/'            Isotopic Compositions          ','Additive BN 200
     * Fractionation Factors (in per mil)'/                              BN 210
      DATA TITLE2/'isotopic composition         ','additive fractionatio BN 220
     *n factor'/                                                         BN 230
      DATA LABEL/'Carbon-13','Carbon-14','Carbon-13','Carbon-14'/        BN 240
      DATA NAME/'CO2','CO2','CH4','CH4'/                                 BN 250
      DATA SULF/'sulfide','sulfate'/                                     BN 260
      ITIME=0                                                            BN 270
   10 CALL CLS                                                           BN 280
      INUM=0                                                             BN 290
      ICO2=0                                                             BN 300
      I2=0                                                               BN 310
      I3=0                                                               BN 320
      DO 140 I=1,NOPHA                                                   BN 330
        IF (ITIME.EQ.1.AND.IT(I).EQ.'+') GO TO 140                       BN 340
        IONE=0                                                           BN 350
        LINE=' '                                                         BN 360
        LINE2=' '                                                        BN 370
        IF (PHASE(I).EQ.'CO2-CH4 ') THEN                                 BN 380
          IONE=1                                                         BN 390
          GO TO 80                                                       BN 400
        END IF                                                           BN 410
        J=0                                                              BN 420
   20   J=J+1                                                            BN 430
        IF (JELE(I,J).EQ.0) GO TO 80                                     BN 440
        IF (JELE(I,J).NE.1.AND.JELE(I,J).NE.2.AND.JELE(I,J).NE.15) GO    BN 450
     *   TO 20                                                           BN 460
        IONE=1                                                           BN 470
        JCO2=1                                                           BN 480
        IF (PHASE(I)(1:3).EQ.'CO2') JCO2=2                               BN 490
        IF (JELE(I,J).EQ.1) THEN                                         BN 500
          WRITE (LINE,40) LINE(1:18),PARA(I,2+4*ITIME),PARA(I,3+4*ITIME) BN 510
     *     ,LINE(43:71)                                                  BN 520
          IF (DOWEHAVE(I,2+4*ITIME)) GO TO 30                            BN 530
          WRITE (LINE,60) LINE(1:18)//' Undefined',LINE(29:71)           BN 540
          CALL CFRACT (FRACTION,JCO2,1D0-0.5D0*IFLAG(1),0.5D0*IFLAG(1),  BN 550
     *     IERROR)                                                       BN 560
          IF (ITIME.EQ.1.AND.ISTATE(I,1).EQ.4) THEN                      BN 570
            IF (IERROR.EQ.0) WRITE (LINE,70) LINE(1:18),FRACTION,'*'//   BN 580
     *       LINE(30:71)                                                 BN 590
            IF (IERROR.EQ.2) WRITE (LINE,70) LINE(1:18),FRACTION,'#'//   BN 600
     *       LINE(30:71)                                                 BN 610
            IF (IERROR.EQ.2) I2=1                                        BN 620
            IF (IERROR.EQ.3) WRITE (LINE,70) LINE(1:18),FRACTION,'@'//   BN 630
     *       LINE(30:71)                                                 BN 640
            IF (IERROR.EQ.3) I3=1                                        BN 650
          END IF                                                         BN 660
   30     IF (DOWEHAVE(I,3+4*ITIME)) GO TO 50                            BN 670
          WRITE (LINE,60) LINE(1:32)//' Undefined',LINE(43:71)           BN 680
          CALL CFRACT (FRACTION,JCO2,1D0-0.5D0*IFLAG(1),0.5D0*IFLAG(1),  BN 690
     *     IERROR)                                                       BN 700
          IF (ITIME.EQ.1.AND.ISTATE(I,1).EQ.4) THEN                      BN 710
            IF (IERROR.EQ.0) WRITE (LINE,70) LINE(1:32),FRACTION*2,'*'// BN 720
```

```
  *       LINE(44:71)                                                  BN 730
         IF (IERROR.EQ.2) WRITE (LINE,70) LINE(1:32),FRACTION*2,'#'// BN 740
  *       LINE(44:71)                                                  BN 750
         IF (IERROR.EQ.2) I2=1                                         BN 760
         IF (IERROR.EQ.3) WRITE (LINE,70) LINE(1:32),FRACTION*2,'@'// BN 770
  *       LINE(44:71)                                                  BN 780
         IF (IERROR.EQ.3) I3=1                                         BN 790
       END IF                                                          BN 800
     END IF                                                            BN 810
40   FORMAT (A,F10.4,4X,F10.4,A)                                       BN 820
50   IF (JELE(I,J).EQ.2) THEN                                          BN 830
       WRITE (LINE,70) LINE(1:46),PARA(I,4+4*ITIME),LINE(57:71)        BN 840
       IF ((ISTATE(I,2).EQ.-1.OR.ISTATE(I,2).EQ.-2).AND.ITIME.EQ.1)    BN 850
  *     THEN                                                           BN 860
         CALL SFRACT (FRACTION,1,1D0-0.5D0*IFLAG(1),0.5D0*IFLAG(1),    BN 870
  *       IERROR)                                                      BN 880
         IF (IERROR.NE.1) THEN                                         BN 890
           IF (.NOT.DOWEHAVE(I,8)) THEN                                BN 900
             CALL HAVE (I,8)                                           BN 910
             PARA(I,8)=0.                                              BN 920
           END IF                                                      BN 930
           IF (IERROR.EQ.0) WRITE (LINE,70) LINE(1:46),PARA(I,8)+      BN 940
  *         FRACTION,'*'//LINE(58:71)                                  BN 950
           IF (IERROR.EQ.2) WRITE (LINE,70) LINE(1:46),PARA(I,8)+      BN 960
  *         FRACTION,'#'//LINE(58:71)                                  BN 970
           IF (IERROR.EQ.2) I2=1                                       BN 980
           IF (IERROR.EQ.3) I3=1                                       BN 990
           IF (IERROR.EQ.3) WRITE (LINE,70) LINE(1:46),PARA(I,8)+      BN1000
  *         FRACTION,'@'//LINE(58:71)                                  BN1010
         ELSE                                                          BN1020
           IF (DOWEHAVE(I,8)) THEN                                     BN1030
             WRITE (LINE,70) LINE(1:46),PARA(I,8),LINE(57:71)          BN1040
           ELSE                                                        BN1050
             WRITE (LINE,60) LINE(1:46),' Undefined ',LINE(58:71)      BN1060
           END IF                                                      BN1070
         END IF                                                        BN1080
       END IF                                                          BN1090
       IF (ISTATE(I,2).EQ.6.AND.ITIME.EQ.1) THEN                       BN1100
         CALL SFRACT (FRACTION,2,1D0-0.5D0*IFLAG(1),0.5D0*IFLAG(1),    BN1110
  *       IERROR)                                                      BN1120
         IF (IERROR.NE.1) THEN                                         BN1130
           IF (.NOT.DOWEHAVE(I,8)) THEN                                BN1140
             CALL HAVE (I,8)                                           BN1150
             PARA(I,8)=0.                                              BN1160
           END IF                                                      BN1170
           IF (IERROR.EQ.0) WRITE (LINE,70) LINE(1:46),PARA(I,8)+      BN1180
  *         FRACTION,'*'//LINE(58:71)                                  BN1190
           IF (IERROR.EQ.2) WRITE (LINE,70) LINE(1:46),PARA(I,8)+      BN1200
  *         FRACTION,'#'//LINE(58:71)                                  BN1210
           IF (IERROR.EQ.2) I2=1                                       BN1220
           IF (IERROR.EQ.3) I3=1                                       BN1230
           IF (IERROR.EQ.3) WRITE (LINE,70) LINE(1:46),PARA(I,8)+      BN1240
  *         FRACTION,'@'//LINE(58:71)                                  BN1250
         ELSE                                                          BN1260
           IF (DOWEHAVE(I,8)) THEN                                     BN1270
             WRITE (LINE,70) LINE(1:46),PARA(I,8),LINE(57:71)          BN1280
           ELSE                                                        BN1290
             WRITE (LINE,60) LINE(1:46),' Undefined ',LINE(58:71)      BN1300
           END IF                                                      BN1310
         END IF                                                        BN1320
       END IF                                                          BN1330
```

```
         IF (.NOT.DOWEHAVE(I,4+4*ITIME)) WRITE (LINE,60) LINE(1:46),' U BN1340
    *ndefined',LINE(57:71)                                              BN1350
       END IF                                                           BN1360
       IF (JELE(I,J).EQ.15) THEN                                        BN1370
         WRITE (LINE,70) LINE(1:60),PARA(I,5+4*ITIME)                   BN1380
         IF (.NOT.DOWEHAVE(I,5+4*ITIME)) THEN                           BN1390
           IF (ITIME.EQ.1) THEN                                         BN1400
             CALL HAVE (I,9)                                            BN1410
             PARA(I,9)=0.                                               BN1420
             WRITE (LINE,70) LINE(1:60),PARA(I,9)                       BN1430
           ELSE                                                         BN1440
             WRITE (LINE,60) LINE(1:60),' Undefined'                    BN1450
           END IF                                                       BN1460
         END IF                                                         BN1470
       END IF                                                           BN1480
  60   FORMAT (A,A,A,A)                                                 BN1490
  70   FORMAT (A,F10.4,A)                                               BN1500
       GO TO 20                                                         BN1510
  80   IF (IONE.EQ.0) GO TO 140                                         BN1520
       INUM=INUM+1                                                      BN1530
       ICCC(INUM)=I                                                     BN1540
       IF (INUM.EQ.1.AND.ITIME.EQ.0) WRITE (*,90) TITLE(0)              BN1550
  90   FORMAT (14X,A43,//,' Number  Phase      Carbon-13   ','C-14 (% m BN1560
    *od)    Sulfur-34   Strontium-87',/,1X,71('-'))                     BN1570
       IF (INUM.EQ.1.AND.ITIME.EQ.1) WRITE (*,100) TITLE(1)             BN1580
 100   FORMAT (14X,A43,/,25X,'Relative to solution',/,' Number  Phase   BN1590
    *    Carbon-13   ','  Carbon-14    Sulfur-34   Strontium-87',/,1X,  BN1600
    *   71('-'))                                                        BN1610
       IF (PHASE(I).EQ.'CO2-CH4 ') GO TO 120                            BN1620
       WRITE (LINE,110) INUM,PHASE(I),LINE(16:71)                       BN1630
 110   FORMAT (I4,':  ',A8,A)                                           BN1640
       WRITE (*,440) LINE(1:LENS(LINE))                                 BN1650
       GO TO 140                                                        BN1660
 120   ICO2=1                                                           BN1670
       WRITE (LINE,110) INUM,'CO2     '                                 BN1680
       WRITE (LINE2,430) '      CH4'                                    BN1690
       IF (ITIME.EQ.1) GO TO 130                                        BN1700
       IF (DOWEHAVE(I,2)) THEN                                          BN1710
         WRITE (LINE,70) LINE(1:18),PARA(I,2),LINE(29:71)               BN1720
       ELSE                                                             BN1730
         WRITE (LINE,60) LINE(1:18),' Undefined',LINE(29:71)            BN1740
       END IF                                                           BN1750
       IF (DOWEHAVE(I,3)) THEN                                          BN1760
         WRITE (LINE,70) LINE(1:32),PARA(I,3),LINE(43:71)               BN1770
       ELSE                                                             BN1780
         WRITE (LINE,60) LINE(1:32),' Undefined',LINE(43:71)            BN1790
       END IF                                                           BN1800
       IF (DOWEHAVE(I,4)) THEN                                          BN1810
         WRITE (LINE2,70) LINE2(1:18),PARA(I,4),LINE2(29:71)            BN1820
       ELSE                                                             BN1830
         WRITE (LINE2,60) LINE2(1:18),' Undefined',LINE2(29:71)         BN1840
       END IF                                                           BN1850
       IF (DOWEHAVE(I,5)) THEN                                          BN1860
         WRITE (LINE2,70) LINE2(1:32),PARA(I,5),LINE2(43:71)            BN1870
       ELSE                                                             BN1880
         WRITE (LINE2,60) LINE2(1:32),' Undefined',LINE2(43:71)         BN1890
```

```
      END IF                                                            BN1900
      WRITE (*,440) LINE(1:LENS(LINE))                                  BN1910
      WRITE (*,440) LINE2(1:LENS(LINE2))                                BN1920
      GO TO 140                                                         BN1930
130   WRITE (LINE,110) INUM,'CO2     '                                  BN1940
      WRITE (LINE2,430) '       CH4'                                    BN1950
      CALL CFRACT (AA,2,1D0-IFLAG(1)*0.5D0,IFLAG(1)*0.5D0,IERROR)       BN1960
      IF (DOWEHAVE(I,6)) THEN                                           BN1970
        WRITE (LINE,70) LINE(1:18),PARA(I,6),LINE(29:71)                BN1980
      ELSE                                                              BN1990
        IF (IERROR.EQ.0) WRITE (LINE,70) LINE(1:18),AA,'*'//            BN2000
   *     LINE(30:71)                                                    BN2010
        IF (IERROR.EQ.1) WRITE (LINE,60) LINE(1:18),' Undefined',       BN2020
   *     LINE(29:71)                                                    BN2030
        IF (IERROR.EQ.2) WRITE (LINE,70) LINE(1:18),AA,'#'//            BN2040
   *     LINE(30:71)                                                    BN2050
        IF (IERROR.EQ.2) I2=1                                           BN2060
        IF (IERROR.EQ.3) I3=1                                           BN2070
        IF (IERROR.EQ.3) WRITE (LINE,70) LINE(1:18),AA,'@'//            BN2080
   *     LINE(30:71)                                                    BN2090
      END IF                                                            BN2100
      IF (DOWEHAVE(I,7)) THEN                                           BN2110
        WRITE (LINE,70) LINE(1:32),PARA(I,7),LINE(43:71)                BN2120
      ELSE                                                              BN2130
        IF (IERROR.EQ.0) WRITE (LINE,70) LINE(1:32),2*AA,'*'//          BN2140
   *     LINE(44:71)                                                    BN2150
        IF (IERROR.EQ.1) WRITE (LINE,60) LINE(1:32),' Undefined',       BN2160
   *     LINE(44:71)                                                    BN2170
        IF (IERROR.EQ.2) WRITE (LINE,70) LINE(1:32),2*AA,'#'//          BN2180
   *     LINE(44:71)                                                    BN2190
        IF (IERROR.EQ.2) I2=1                                           BN2200
        IF (IERROR.EQ.3) I3=1                                           BN2210
        IF (IERROR.EQ.3) WRITE (LINE,70) LINE(1:32),2*AA,'@'//          BN2220
   *     LINE(44:71)                                                    BN2230
      END IF                                                            BN2240
      IF (DOWEHAVE(I,8)) THEN                                           BN2250
        WRITE (LINE2,70) LINE2(1:18),PARA(I,8),LINE2(29:71)             BN2260
      ELSE                                                              BN2270
        WRITE (LINE2,60) LINE2(1:18),' Undefined',LINE2(29:71)          BN2280
      END IF                                                            BN2290
      IF (DOWEHAVE(I,9)) THEN                                           BN2300
        WRITE (LINE2,70) LINE2(1:32),PARA(I,9),LINE2(43:71)             BN2310
      ELSE                                                              BN2320
        WRITE (LINE2,60) LINE2(1:32),' Undefined',LINE2(43:71)          BN2330
      END IF                                                            BN2340
      WRITE (*,440) LINE(1:LENS(LINE))                                  BN2350
      WRITE (*,440) LINE2(1:LENS(LINE2))                                BN2360
140 CONTINUE                                                            BN2370
    IF (INUM.EQ.0) THEN                                                 BN2380
      RETURN                                                            BN2390
    END IF                                                              BN2400
150 CALL POSCUR (INUM+ICO2)                                             BN2410
    CALL CLPART                                                         BN2420
    IF (ITIME.EQ.0) THEN                                                BN2430
      WRITE (*,160)                                                     BN2440
    ELSE                                                                BN2450
      WRITE (*,170) DISALONG                                            BN2460
      IF (I2.EQ.1) WRITE (*,180)                                        BN2470
      IF (I3.EQ.1) WRITE (*,190)                                        BN2480
      WRITE (*,200)                                                     BN2490
    END IF                                                              BN2500
```

```
160 FORMAT (/,' Enter number of phase to edit. <Enter> when done.')
170 FORMAT (' * = based on computed value at ',F4.2,' fraction',' betw
   *een init and final waters')
180 FORMAT (' # = based on computed value at initial water')
190 FORMAT (' @ = based on computed value at final water')
200 FORMAT (/' Enter number of phase to edit, ''0'' for fraction',' al
   *ong path, <Enter> when done.')
    READ (*,430,ERR=150) LINE
    IF (LINE.EQ.' ') THEN
      IF (ITIME.EQ.1) THEN
        RETURN
      END IF
      ITIME=1
      CALL ICCARBON
      GO TO 10
    END IF
    READ (LINE,'(I5)',ERR=150) I
    IF (I.EQ.0.AND.ITIME.EQ.1) GO TO 410
    IF (I.LT.1.OR.I.GT.INUM) GO TO 150
    IA1=0
    IA2=0
    IA3=0
    J=0
210 J=J+1
    IF (JELE(ICCC(I),J).EQ.0) GO TO 220
    IF (JELE(ICCC(I),J).EQ.1) IA1=1
    IF (JELE(ICCC(I),J).EQ.2) IA2=1
    IF (JELE(ICCC(I),J).EQ.15) IA3=1
    GO TO 210
220 IF (IA1.EQ.0) GO TO 290
    IWHICH=1
230 CALL POSCUR (INUM+ICO2+1+ITIME)
    CALL CLPART
    IF (ITIME.EQ.0) GO TO 240
    IF (PHASE(ICCC(I))(1:3).EQ.'CO2'.AND.IWHICH.LE.2) THEN
      CALL CFRACT (FRACT1,2,1D0,0D0,IERROR)
      CALL CFRACT (FRACT2,2,0D0,1D0,IERROR)
      IF (IFLAG(1).EQ.0) THEN
        WRITE (*,370) FRACT1*IWHICH
      ELSE
        WRITE (*,380) FRACT1*IWHICH,FRACT2*IWHICH
      END IF
    ELSE IF (ISTATE(ICCC(I),1).EQ.4) THEN
      CALL CFRACT (FRACT1,1,1D0,0D0,IERROR)
      CALL CFRACT (FRACT2,1,0D0,1D0,IERROR)
      IF (IFLAG(1).EQ.0) THEN
        WRITE (*,390) FRACT1*IWHICH
      ELSE
        WRITE (*,400) FRACT1*IWHICH,FRACT2*IWHICH
      END IF
    END IF
240 IF (PHASE(ICCC(I)).EQ.'CO2-CH4 ') THEN
      WRITE (*,360) TITLE2(ITIME)(1:LENS(TITLE2(ITIME))),LABEL(IWHICH)
   *    ,NAME(IWHICH)
    ELSE
      WRITE (*,360) TITLE2(ITIME)(1:LENS(TITLE2(ITIME))),LABEL(IWHICH)
   *    ,PHASE(ICCC(I))
    END IF
    IF ((ISTATE(ICCC(I),1).EQ.4.AND.ITIME.EQ.1).OR.(PHASE(ICCC(I)).EQ.
   *'CO2-CH4 '.AND.ITIME.EQ.1.AND.IWHICH.LT.3)) THEN
      IF (DOWEHAVE(ICCC(I),1+4*ITIME+IWHICH)) THEN
```

```
          WRITE (*,250) PARA(ICCC(I),1+4*ITIME+IWHICH)
        ELSE
          WRITE (*,260)
        END IF
      ELSE
        IF (DOWEHAVE(ICCC(I),1+4*ITIME+IWHICH)) WRITE (*,270)
     *   PARA(ICCC(I),1+4*ITIME+IWHICH)
      END IF
  250 FORMAT (' Type ''CALC'' to use calculated value or <Enter> for',
     *F10.4)
  260 FORMAT (' <Enter> to use calculated value.')
  270 FORMAT (' <Enter> to use',F10.4)
      READ (*,430) LINE
      IF (LINE.EQ.' '.AND.PHASE(ICCC(I)).EQ.'CO2-CH4 '.AND.IWHICH.LT.3)
     *GO TO 280
      IF (LINE.EQ.' '.AND..NOT.DOWEHAVE(ICCC(I),1+4*ITIME+IWHICH).AND.
     *ISTATE(ICCC(I),1).NE.4) THEN
        CALL POSCUR (INUM+ICO2+1+ITIME)
        CALL CLPART
        GO TO 240
      END IF
      IF (LINE.EQ.' '.AND..NOT.DOWEHAVE(ICCC(I),1+4*ITIME+IWHICH).AND.
     *ITIME.EQ.0) THEN
        CALL POSCUR (INUM+ICO2+1+ITIME)
        CALL CLPART
        GO TO 240
      END IF
      IF (LINE.EQ.' ') GO TO 280
      IF (UPCS(LINE(1:4)).EQ.'CALC'.AND.ISTATE(ICCC(I),1).EQ.4.AND.
     *ITIME.EQ.1) THEN
        CALL DONTHAVE (ICCC(I),1+4*ITIME+IWHICH)
        GO TO 280
      END IF
      IF (UPCS(LINE(1:4)).EQ.'CALC'.AND.PHASE(ICCC(I)).EQ.'CO2-CH4 '.
     *AND.ITIME.EQ.1.AND.IWHICH.LT.3) THEN
        CALL DONTHAVE (ICCC(I),1+4*ITIME+IWHICH)
        GO TO 280
      END IF
      READ (LINE,'(F20.0)',ERR=240) CTH
      IF (IWHICH.EQ.2.AND.CTH.LT.0) GO TO 240
      IF (IWHICH.NE.2.AND.CTH.LT.-1D3) GO TO 240
      PARA(ICCC(I),1+4*ITIME+IWHICH)=CTH
      CALL HAVE (ICCC(I),1+4*ITIME+IWHICH)
  280 IWHICH=IWHICH+1
      IF (IWHICH.LT.3) GO TO 230
      IF (IWHICH.LT.5.AND.PHASE(ICCC(I)).EQ.'CO2-CH4') GO TO 230
C
C SULFUR EDITING
  290 IF (IA2.EQ.0) GO TO 340
      CALL POSCUR (INUM+ICO2+1+ITIME)
      CALL CLPART
      IST=ISTATE(ICCC(I),2)
      ISULF=0
      IF (ITIME.EQ.1.AND.(IST.EQ.-1.OR.IST.EQ.-2).AND.NODATA(WELL(3),23)
     *.EQ.0.AND.DBDATA(WELL(3),2).GT.0..AND.DBDATA(WELL(3),29).GT.0.)
     *ISULF=1
      IF (ITIME.EQ.1.AND.IST.EQ.6.AND.NODATA(WELL(3),23).EQ.0.AND.
     *DBDATA(WELL(3),2).GT.0..AND.DBDATA(WELL(3),30).GT.0.) ISULF=2
      IF (ISULF.GT.0) THEN
        CALL SFRACT (FRACT1,ISULF,1D0,0D0,IERROR)
        IF (IFLAG(1).EQ.1) CALL SFRACT (FRACT2,ISULF,0D0,1D0,IERROR)
```

```
        IF (IERROR.EQ.0) THEN                                            BN3730
          IF (.NOT.DOWEHAVE(ICCC(I),8)) PARA(ICCC(I),8)=0.               BN3740
          CALL HAVE (ICCC(I),8)                                          BN3750
          IF (IFLAG(1).EQ.0) THEN                                        BN3760
            WRITE (*,300) SULF(ISULF),FRACT1                             BN3770
          ELSE                                                           BN3780
            WRITE (*,310) SULF(ISULF),FRACT1,FRACT2                      BN3790
          END IF                                                         BN3800
        ELSE                                                             BN3810
          ISULF=0                                                        BN3820
        END IF                                                           BN3830
      END IF                                                             BN3840
  300 FORMAT (' The calculated fractionation factor between ',A,' and',' BN3850
     * total sulfur is',/,F11.4)                                         BN3860
  310 FORMAT (' The calculated fractionation factor between ',A,' and',' BN3870
     * total sulfur is',/,' between ',F10.4,' and ',F10.4)               BN3880
  320 IF (DOWEHAVE(ICCC(I),4+4*ITIME)) THEN                              BN3890
        WRITE (*,360) TITLE2(ITIME)(1:LENS(TITLE2(ITIME))),'Sulfur-34',  BN3900
     *   PHASE(ICCC(I)),PARA(ICCC(I),4+4*ITIME)                          BN3910
      ELSE                                                               BN3920
        WRITE (*,360) TITLE2(ITIME)(1:LENS(TITLE2(ITIME))),'Sulfur-34',  BN3930
     *   PHASE(ICCC(I))                                                  BN3940
      END IF                                                             BN3950
      IF (ISULF.GT.0) WRITE (*,330) SULF(ISULF)                          BN3960
  330 FORMAT (' (Value will be added to ',A,'-sulfur value)')            BN3970
      READ (*,430) LINE                                                  BN3980
      IF (LINE.NE.' ') THEN                                              BN3990
        READ (LINE,'(f20.0)',ERR=320) CTH                                BN4000
        IF (CTH.LT.-1D3) GO TO 320                                       BN4010
        PARA(ICCC(I),4+4*ITIME)=CTH                                      BN4020
      ELSE                                                               BN4030
        IF (.NOT.DOWEHAVE(ICCC(I),4+4*ITIME)) THEN                       BN4040
          CALL POSCUR (INUM+ICO2+1+ITIME)                                BN4050
          CALL CLPART                                                    BN4060
          GO TO 320                                                      BN4070
        END IF                                                           BN4080
      END IF                                                             BN4090
      CALL HAVE (ICCC(I),4+4*ITIME)                                      BN4100
  340 IF (IA3.EQ.0) GO TO 10                                             BN4110
  350 CALL POSCUR (INUM+ICO2+1+ITIME)                                    BN4120
      CALL CLPART                                                        BN4130
      IF (DOWEHAVE(ICCC(I),5+4*ITIME)) THEN                              BN4140
        WRITE (*,360) TITLE2(ITIME)(1:LENS(TITLE2(ITIME))),'Strontium-87 BN4150
     *',PHASE(ICCC(I)),PARA(ICCC(I),5+4*ITIME)                           BN4160
      ELSE                                                               BN4170
        WRITE (*,360) TITLE2(ITIME)(1:LENS(TITLE2(ITIME))),'Strontium-87 BN4180
     *',PHASE(ICCC(I))                                                   BN4190
      END IF                                                             BN4200
      READ (*,430) LINE                                                  BN4210
      IF (LINE.NE.' ') THEN                                              BN4220
        READ (LINE,'(f20.0)',ERR=350) CTH                                BN4230
        IF (CTH.LT.-1D3) GO TO 350                                       BN4240
        PARA(ICCC(I),5+4*ITIME)=CTH                                      BN4250
      ELSE                                                               BN4260
        IF (.NOT.DOWEHAVE(ICCC(I),5+4*ITIME)) THEN                       BN4270
          CALL POSCUR (INUM+ICO2+1+ITIME)                                BN4280
          CALL CLPART                                                    BN4290
          GO TO 320                                                      BN4300
        END IF                                                           BN4310
      END IF                                                             BN4320
      CALL HAVE (ICCC(I),5+4*ITIME)                                      BN4330
```

```
  360 FORMAT (' Enter ',A,' of ',A,' for ''',A,'''.',:,/' <Enter> for ', BN4340
     *F10.4)                                                            BN4350
  370 FORMAT (' The calculated additive fractionation factor for CO2 is' BN4360
     *,/,F11.4)                                                         BN4370
  380 FORMAT (' The calculated additive fractionation factor for CO2 is' BN4380
     *,/,' between',F10.4,' and',F10.4)                                 BN4390
  390 FORMAT (' The calculated additive fractionation factor for',' carb BN4400
     *onates is',/,F11.4)                                               BN4410
  400 FORMAT (' The calculated additive fractionation factor for',' carb BN4420
     *onates is',/,' between',F10.4,' and',F10.4)                       BN4430
      GO TO 10                                                          BN4440
  410 CALL POSCUR (INUM+ICO2+1+ITIME)                                   BN4450
      CALL CLPART                                                       BN4460
      WRITE (*,420) DISALONG                                            BN4470
  420 FORMAT (' Enter fractional distance along path (0=initial,1=final) BN4480
     *',/,' <Enter> for ',F5.3,'.')                                     BN4490
      READ (*,430) LINE                                                 BN4500
  430 FORMAT (A)                                                        BN4510
  440 FORMAT (1X,A)                                                     BN4520
      IF (LINE.EQ.' ') GO TO 150                                        BN4530
      IF (LINE(1:1).NE.'0'.AND.LINE(1:1).NE.'.'.AND.LINE(1:1).NE.'1')   BN4540
     *GO TO 410                                                         BN4550
      READ (LINE,'(F20.0)',ERR=410) CTH                                 BN4560
      IF (CTH.LT.0.0.OR.CTH.GT.1.0) GO TO 410                           BN4570
      DISALONG=CTH                                                      BN4580
      GO TO 10                                                          BN4590
      END                                                               BN4600
      INTEGER FUNCTION ISTATE(I,J)                                      BO  10
C                                                                       BO  20
C The redox state of either carbon or sulfur in a given phase is        BO  30
C calculated.  If the list of constraints is changed, this subroutine   BO  40
C may have to be modified.                                              BO  50
C                                                                       BO  60
$INCLUDE:'NETPATH.BLO'                                                  BO  70
      IRS=0                                                             BO  80
      K=0                                                               BO  90
   10 K=K+1                                                             BO 100
      IF (JELE(I,K).EQ.0) GO TO 20                                      BO 110
      IF (JELE(I,K).NE.20) GO TO 10                                     BO 120
      IRS=PCOEFF(I,K)                                                   BO 130
   20 ISTATE=0                                                          BO 140
      IDIV=1                                                            BO 150
      K=0                                                               BO 160
   30 K=K+1                                                             BO 170
      IF (JELE(I,K).EQ.0) GO TO 40                                      BO 180
      IF (JELE(I,K).EQ.1.AND.J.NE.1) ISTATE=ISTATE+4*PCOEFF(I,K)        BO 190
      IF (JELE(I,K).EQ.2.AND.J.NE.2) ISTATE=ISTATE+6*PCOEFF(I,K)        BO 200
      IF (JELE(I,K).EQ.16) ISTATE=ISTATE+2*PCOEFF(I,K)                  BO 210
      IF (JELE(I,K).EQ.17) ISTATE=ISTATE+2*PCOEFF(I,K)                  BO 220
      IF (JELE(I,K).EQ.J) IDIV=PCOEFF(I,K)                              BO 230
      GO TO 30                                                          BO 240
   40 ISTATE=(IRS-ISTATE)/IDIV                                          BO 250
      RETURN                                                            BO 260
      END                                                               BO 270
      INTEGER FUNCTION LENS(STRING)                                     BP  10
C                                                                       BP  20
C The length of a string, not including trailing blanks, is returned.   BP  30
C This is used to improve screen prints.                                BP  40
C                                                                       BP  50
      CHARACTER*(*) STRING                                              BP  60
      K=LEN(STRING)                                                     BP  70
```

```
      LENS=0
      DO 10 I=K,1,-1
        IF (STRING(I:I).NE.' '.AND.LENS.EQ.0) LENS=I
   10 CONTINUE
      RETURN
      END
      SUBROUTINE LISTCON (NELE)
C
C The constraints currently under consideration are listed.  This is
C different from CONLIST, which lists all constraints that the program
C uses.
C
$INCLUDE:'NETPATH.BLO'
      CALL CLS
      WRITE (*,10)
   10 FORMAT (/' List of constraints',/)
      I=1
   20 JJ=I+1
      KK=JJ+1
      IF (KK.EQ.NELE) GO TO 60
      IF (JJ.EQ.NELE) GO TO 40
      IF (I.NE.NELE) GO TO 60
      WRITE (*,30) I,ELELONG(IELE(I))
   30 FORMAT (I4,': ',A14)
      GO TO 80
   40 WRITE (*,50) I,ELELONG(IELE(I)),JJ,ELELONG(IELE(JJ))
   50 FORMAT (I4,': ',A14,8X,I3,': ',A14)
      GO TO 80
   60 WRITE (*,70) I,ELELONG(IELE(I)),JJ,ELELONG(IELE(JJ)),KK,
     *ELELONG(IELE(KK))
   70 FORMAT (I4,': ',A14,8X,I3,': ',A14,8X,I3,': ',A14)
   80 IF (I.EQ.NELE.OR.I+1.EQ.NELE.OR.I+2.EQ.NELE) RETURN
      I=I+3
      GO TO 20
      END
      SUBROUTINE LISTPHA (II)
C
C The phases currently included in the model are listed.  This is differ
C from PHALIST, which lists all the phases available from NETPATH.DAT.
C
$INCLUDE:'NETPATH.BLO'
      II=0
      CALL CLS
      WRITE (*,10)
   10 FORMAT (//,' List of phases',/)
      I=1
      IJ=1
   20 JJ=I+1
      KK=JJ+1
      IF (I.LT.48*IJ) GO TO 50
      WRITE (*,30)
   30 FORMAT (' Hit RETURN to see next page or enter number of phase.')
      IJ=IJ+1
      READ (*,40) BANS
   40 FORMAT (A)
      IF (BANS.EQ.' ') GO TO 50
      READ (BANS,'(i3)',ERR=50) II
      RETURN
   50 IF (KK.EQ.NOPHA) GO TO 100
      IF (JJ.EQ.NOPHA) GO TO 90
      IF (I.NE.NOPHA) GO TO 100
```

```
      WRITE (*,60) I,PHASE(I)
   60 FORMAT (I4,': ',A10)
   70 FORMAT (I4,': ',A10,10X,I3,': ',A10)
   80 FORMAT (I4,': ',A10,10X,I3,': ',A10,10X,I3,': ',A10)
      GO TO 110
   90 WRITE (*,70) I,PHASE(I),JJ,PHASE(JJ)
      GO TO 110
  100 WRITE (*,80) I,PHASE(I),JJ,PHASE(JJ),KK,PHASE(KK)
  110 IF (I.EQ.NOPHA.OR.I+1.EQ.NOPHA.OR.I+2.EQ.NOPHA) RETURN
      I=I+3
      GO TO 20
      END
      SUBROUTINE MODELS
C
C The previously stored models for the given well file are displayed,
C one may be selected, and if so the data from it are read in.
C
$INCLUDE:'NETPATH.BLO'
      CHARACTER*7 DEL(0:1)
      CHARACTER*2 SHORT,PELEM(15)
      CHARACTER*8 PHASES(39),MIDDLE
      CHARACTER*10 MID
      CHARACTER*80 FILES(100),OFILES(100)
      INTEGER IDEL(100),IELEM(39)
      DATA DEL/'       ','deleted'/
      IF (IEDIT.EQ.0) NOELE=0
      I=0
      CLOSE (7)
      OPEN (UNIT=7,FILE='MODEL.FIL',STATUS='OLD',ERR=530)
      REWIND (7)
      NFILES=0
      NOFILES=0
      READ (7,10,ERR=30,END=530) LINE
   10 FORMAT (A80)
      IF (LINE.EQ.' ') GO TO 30
   20 OPEN (8,FILE=LINE,STATUS='OLD',ERR=30)
      READ (8,10,ERR=30,END=30) PFILE
      CLOSE (8)
      IF (PFILE.NE.WFILE) GO TO 40
      NFILES=NFILES+1
      FILES(NFILES)=LINE
      IDEL(NFILES)=0
   30 READ (7,10,ERR=30,END=50) LINE
      GO TO 20
   40 NOFILES=NOFILES+1
      OFILES(NOFILES)=LINE
      GO TO 30
   50 IF (NFILES.EQ.0) GO TO 500
      IJ=1
   60 ICOUNT=(IJ-1)*15
      CALL CLS
      WRITE (*,70)
   70 FORMAT (/,' ------------------',/,' CHOOSE MODEL FILE',/,' --------
     *----------',/)
   80 ICOUNT=ICOUNT+1
      WRITE (*,90) ICOUNT,FILES(ICOUNT),DEL(IDEL(ICOUNT))
   90 FORMAT (I4,':  ',A40,A7)
      IF (ICOUNT.LT.NFILES.AND.ICOUNT.LT.IJ*15) GO TO 80
      DO 100 I=ICOUNT,(IJ*15-1)
        WRITE (*,*)
  100 CONTINUE
```

```
      IF (IJ*15.LT.NFILES) THEN                                         BS 500
        IJ=IJ+1                                                         BS 510
      ELSE                                                              BS 520
        IJ=1                                                            BS 530
      END IF                                                            BS 540
      IF (IEDIT.EQ.0) THEN                                              BS 550
        IF (NFILES.LE.15) WRITE (*,120)                                 BS 560
        IF (NFILES.GT.15) WRITE (*,110)                                 BS 570
        NOELE=0                                                         BS 580
      ELSE                                                              BS 590
        IF (NFILES.LE.15) WRITE (*,140)                                 BS 600
        IF (NFILES.GT.15) WRITE (*,130)                                 BS 610
      END IF                                                            BS 620
  110 FORMAT (/,' Enter number of file, 0 for none, or <ENTER> to see',' BS 630
     * more files:')                                                    BS 640
  120 FORMAT (/,' Enter number of file or 0 for none:')                 BS 650
  130 FORMAT (/,' Enter number of file, ''M'' to see more choices, 0 to' BS 660
     *,' reset the model',/,'  or <ENTER> to keep the current model:')  BS 670
  140 FORMAT (/,' Enter number of file, 0 to reset model, or <ENTER>',' BS 680
     *to keep current model:')                                          BS 690
      READ (*,10) LINE                                                  BS 700
      IF (LINE.EQ.' '.AND.IEDIT.EQ.0) GO TO 60                          BS 710
      IF (LINE.EQ.' ') GO TO 500                                        BS 720
      READ (LINE,'(i3)',ERR=60) I                                       BS 730
      IF (I.EQ.0) THEN                                                  BS 740
        CALL INITVALS (1)                                               BS 750
        GO TO 500                                                       BS 760
      END IF                                                            BS 770
      IF (I.GT.NFILES) GO TO 50                                         BS 780
      CALL CLS                                                          BS 790
      NOELEM=0                                                          BS 800
      WRITE (*,150) FILES(I),DEL(IDEL(I))                               BS 810
  150 FORMAT (1X,40('-'),/,1X,A40,A20,/,1X,40('-'))                     BS 820
      OPEN (8,FILE=FILES(I))                                            BS 830
      READ (8,'(1X)',END=300)                                           BS 840
C GET AND PRINT WELLS                                                   BS 850
      READ (8,160,ERR=300,END=300) (WELL(IJ),IJ=1,3)                    BS 860
  160 FORMAT (3I3)                                                      BS 870
      IFLAG(1)=1                                                        BS 880
      IF (WELL(3).EQ.0) THEN                                            BS 890
        IFLAG(1)=0                                                      BS 900
        WELL(3)=WELL(2)                                                 BS 910
        WELL(2)=0                                                       BS 920
      END IF                                                            BS 930
      WELL(1)=FINDTOT(WELL(1))                                          BS 940
      WELL(2)=FINDTOT(WELL(2))                                          BS 950
      WELL(3)=FINDTOT(WELL(3))                                          BS 960
      WRITE (*,170) WLLNMS(WELL(1))(5:36)                               BS 970
      IF (IFLAG(1).EQ.1) WRITE (*,170) WLLNMS(WELL(2))(5:36)            BS 980
      WRITE (*,180) WLLNMS(WELL(3))(5:36)                               BS 990
  170 FORMAT (' Initial well: ',A32)                                    BS1000
  180 FORMAT (' Final well  : ',A32,/)                                  BS1010
  190 FORMAT (A,A)                                                      BS1020
C GET AND PRINT CONSTRAINTS                                             BS1030
      READ (8,190,END=300) LINE                                         BS1040
      IIC=0                                                             BS1050
  200 IIC=IIC+1                                                         BS1060
      READ (LINE(3*IIC-1:3*IIC),210) SHORT                              BS1070
  210 FORMAT (A2)                                                       BS1080
      IF (SHORT.EQ.'  ') GO TO 230                                      BS1090
      J=0                                                               BS1100
```

```
  220 J=J+1                                                              BS1110
      IF (J.EQ.29) GO TO 200                                             BS1120
      IF (ELESHORT(J).NE.SHORT) GO TO 220                                BS1130
      NOELEM=NOELEM+1                                                    BS1140
      IELEM(NOELEM)=J                                                    BS1150
      GO TO 200                                                          BS1160
  230 WRITE (*,240)                                                      BS1170
  240 FORMAT (' CONSTRAINTS:')                                           BS1180
      WRITE (*,250) (ELELONG(IELEM(LI)),LI=1,NOELEM)                     BS1190
  250 FORMAT (:,5(1X,A12,:,2X))                                          BS1200
C                                                                        BS1210
C    GET AND PRINT PHASES                                                BS1220
C                                                                        BS1230
      NOPHAS=0                                                           BS1240
  260 READ (8,'(A10,A2)',END=300) MID,SHORT                              BS1250
      IF (MID.EQ.' '.AND.SHORT.EQ.'  ') GO TO 270                        BS1260
      IF (MID.EQ.' ') GO TO 260                                          BS1270
      NOPHAS=NOPHAS+1                                                    BS1280
      PHASES(NOPHAS)=MID(1:8)                                            BS1290
      GO TO 260                                                          BS1300
  270 WRITE (*,280)                                                      BS1310
  280 FORMAT (/,' PHASES:')                                              BS1320
      WRITE (*,290) (PHASES(LI),LI=1,NOPHAS)                             BS1330
  290 FORMAT (:,5(1X,A8,:,6X))                                           BS1340
C DONE READING DATA                                                      BS1350
  300 CLOSE (8)                                                          BS1360
      IF (IDEL(I).NE.1) WRITE (*,310)                                    BS1370
      IF (IDEL(I).EQ.1) WRITE (*,320)                                    BS1380
  310 FORMAT (/,' Use this file?  (N)o, (D)elete, or <ENTER> to accept:' BS1390
     *)                                                                  BS1400
  320 FORMAT (/,' Undelete this file?  un(D)elete, or <ENTER> for no:')  BS1410
      READ (*,190) ANS                                                   BS1420
      ANS=UPCS(ANS)                                                      BS1430
      IF (IDEL(I).EQ.1.AND.ANS.NE.'D') GO TO 50                          BS1440
      IF (ANS.EQ.'D') THEN                                               BS1450
        IDEL(I)=1-IDEL(I)                                                BS1460
        GO TO 50                                                         BS1470
      END IF                                                             BS1480
      IF (ANS.EQ.'N') GO TO 50                                           BS1490
      IDEL(I)=0                                                          BS1500
C MAKE ELEMENTS PERMANENT                                                BS1510
      NOELE=NOELEM                                                       BS1520
      DO 330 ICT=1,NOELE                                                 BS1530
  330   IELE(ICT)=IELEM(ICT)                                             BS1540
      EFILE=FILES(I)                                                     BS1550
C MAKE PHASES PERMANENT                                                  BS1560
      NOPHA=NOPHAS                                                       BS1570
      DO 340 ICT=1,NOPHA                                                 BS1580
  340   PHASE(ICT)=PHASES(ICT)                                           BS1590
      OPEN (UNIT=8,FILE=FILES(I),STATUS='OLD',ERR=490)                   BS1600
      READ (8,190,END=450) LINE                                          BS1610
      READ (8,190,END=450) LINE                                          BS1620
      READ (8,190,END=450) LINE                                          BS1630
      I=0                                                                BS1640
  350 I=I+1                                                              BS1650
      IF (I.GT.NOPHA) GO TO 470                                          BS1660
      READ (8,360,ERR=450,END=450) MIDDLE,F(I),IT(I),(PELEM(J),PCOEFF(I, BS1670
     *J),J=1,7)                                                          BS1680
      PARA(I,1)=0.                                                       BS1690
      CALL DONTHAVE (I,1)                                                BS1700
  360 FORMAT (A8,2(A1),7(A2,F8.4))                                       BS1710
```

```
      IF (MIDDLE.NE.' ') GO TO 380                                      BS1720
      DO 370 J=1,7                                                      BS1730
        PELEM(J+7)=PELEM(J)                                             BS1740
  370   PCOEFF(I-1,J+7)=PCOEFF(I,J)                                     BS1750
      I=I-1                                                             BS1760
      J=0                                                               BS1770
      JO=7                                                              BS1780
      GO TO 390                                                         BS1790
  380 J=0                                                               BS1800
      JO=0                                                              BS1810
  390 J=J+1                                                             BS1820
      IF (J.EQ.8) THEN                                                  BS1830
        PCOEFF(I,J+JO)=0                                                BS1840
        GO TO 350                                                       BS1850
      END IF                                                            BS1860
      IF (PELEM(J+JO).NE.'  ') GO TO 400                                BS1870
      JELE(I,J+JO)=0                                                    BS1880
      GO TO 350                                                         BS1890
  400 K=0                                                               BS1900
  410 K=K+1                                                             BS1910
      IF (PELEM(J+JO).EQ.ELESHORT(K)) GO TO 430                         BS1920
      IF (K.LT.33) GO TO 410                                            BS1930
  420 WRITE (*,'(//,i3,1x,a2,'' : '',a,1x,i2)') J+JO,PELEM(J+JO),'Bad coBS1940
     *nstraint in phase #',I                                            BS1950
      STOP                                                              BS1960
  430 IF (K.GT.20) GO TO 440                                            BS1970
      JELE(I,J+JO)=K                                                    BS1980
      GO TO 390                                                         BS1990
  440 IF (K.GT.24.AND.K.LT.29) GO TO 420                                BS2000
      IF (K.EQ.33) THEN                                                 BS2010
        IT(I)='*'                                                       BS2020
        PARA(I,1)=PCOEFF(I,J+JO)                                        BS2030
      END IF                                                            BS2040
      IF (K.LE.24) THEN                                                 BS2050
        PARA(I,K-19)=PCOEFF(I,J+JO)                                     BS2060
        CALL HAVE (I,K-19)                                              BS2070
      END IF                                                            BS2080
      IF (K.GE.29.AND.K.LE.32) THEN                                     BS2090
        PARA(I,K-23)=PCOEFF(I,J+JO)                                     BS2100
        CALL HAVE (I,K-23)                                              BS2110
      END IF                                                            BS2120
      JELE(I,J+JO)=0                                                    BS2130
      GO TO 390                                                         BS2140
  450 IF (I.GT.NOPHA) GO TO 470                                         BS2150
      WRITE (*,460) EFILE                                               BS2160
      GO TO 490                                                         BS2170
  460 FORMAT (//' Error in file: ',A)                                   BS2180
C MAKE PARAMETERS PERMANENT                                             BS2190
  470 READ (8,480,ERR=490,END=490) (IFLAG(I),I=2,6),(P(I),I=1,2),       BS2200
     *(ISDOCRS(I),I=1,3),DISALONG,(C14DAT(I),I=1,9),I10,I11,(C14DAT(I), BS2210
     *I=12,13),(DBDATA(WELL(3),I),I=44,47),DBDATA(WELL(3),49),          BS2220
     *(DBDATA(WELL(1),I),I=44,47),DBDATA(WELL(1),49),(DBDATA(WELL(2),I),BS2230
     *I=44,47),DBDATA(WELL(2),49)                                       BS2240
  480 FORMAT (/,5(I2),2(F8.4),3(I2),F6.3,/,7(F8.3),/,2(F8.0),2(I8),2(F8.BS2250
     *0),/,3(5(F8.3),/))                                                BS2260
  490 CLOSE (8)                                                         BS2270
C SAVE MODEL FILE LIST                                                  BS2280
  500 REWIND (UNIT=7)                                                   BS2290
      WRITE (7,'(a80)') ' '                                             BS2300
      REWIND (UNIT=7)                                                   BS2310
      DO 510 J=1,NFILES                                                 BS2320
```

```
        IF (IDEL(J).NE.1) WRITE (7,190) FILES(J)(1:LENS(FILES(J)))      BS2330
  510 CONTINUE                                                          BS2340
      DO 520 J=1,NOFILES                                                BS2350
  520   WRITE (7,190) OFILES(J)                                         BS2360
  530 IF (I.EQ.0.OR.WELL(1).EQ.0.OR.WELL(3).EQ.0.OR.(WELL(2).EQ.0.AND.  BS2370
      *IFLAG(1).EQ.1)) CALL WELLS                                       BS2380
      RETURN                                                            BS2390
      END                                                               BS2400
      SUBROUTINE NEXT                                                   BT  10
$INCLUDE:'NETPATH.BLO'                                                  BT  20
C                                                                       BT  30
C             NOW(I) CONTAINS THE CURRENT MINERAL CHOICES               BT  40
C             IN ARRAY POSITIONS 1 THROUGH NEQ                          BT  50
C                                                                       BT  60
      QUIT=.FALSE.                                                      BT  70
C                                                                       BT  80
C             FIRST SET OF MINERALS IS CHOSEN                           BT  90
C                                                                       BT 100
      IF (NEQ.GT.NMINS) THEN                                            BT 110
        QUIT=.TRUE.                                                     BT 120
        RETURN                                                          BT 130
      END IF                                                            BT 140
      IF (FIRST) THEN                                                   BT 150
        DO 10 I=1,NEQ                                                   BT 160
          MINPOS(I)=I                                                   BT 170
          MAXPOS(I)=NMINS-NEQ+I                                         BT 180
          IF (FORCE(I).EQ.'F') MAXPOS(I)=MINPOS(I)                      BT 190
          NOW(I)=I                                                      BT 200
   10   CONTINUE                                                        BT 210
        IF (IMIX.EQ.1.OR.IFLAG(6).EQ.1) THEN                            BT 220
          MAXPOS(1)=1                                                   BT 230
          MAXPOS(2)=2                                                   BT 240
        END IF                                                          BT 250
        IF (IMIX.EQ.1.AND.IFLAG(6).EQ.1) MAXPOS(3)=3                    BT 260
        FIRST=.FALSE.                                                   BT 270
        RETURN                                                          BT 280
      END IF                                                            BT 290
C                                                                       BT 300
C             SET OF MINERALS (OTHER THAN FIRST TIME)                   BT 310
C             IS CHOSEN                                                 BT 320
      DO 30 I=NEQ,1,-1                                                  BT 330
        IF (NOW(I).LT.MAXPOS(I)) THEN                                   BT 340
          NOW(I)=NOW(I)+1                                               BT 350
          IF (I.LT.NEQ) THEN                                            BT 360
            K=NOW(I)                                                    BT 370
            DO 20 J=I+1,NEQ                                             BT 380
              K=K+1                                                     BT 390
              NOW(J)=K                                                  BT 400
   20       CONTINUE                                                    BT 410
          END IF                                                        BT 420
          RETURN                                                        BT 430
        END IF                                                          BT 440
   30 CONTINUE                                                          BT 450
      QUIT=.TRUE.                                                       BT 460
      RETURN                                                            BT 470
      END                                                               BT 480
      SUBROUTINE PHALIST (II)                                           BU  10
C                                                                       BU  20
C The available phases (from NETPATH.DAT) are displayed.  Also, if      BU  30
C there are more than 60, the display is paused after the first 60, and BU  40
C the next page can be seen or the number of the phase to be used can be BU  50
```

```
C entered.
C
$INCLUDE:'NETPATH.BLO'
      II=0
      WRITE (*,10)
   10 FORMAT (//,' List of phases and their corresponding numbers.')
      IF (FLIN.EQ.0) THEN
        WRITE (*,*) 'No phases - check NETPATH.DAT'
        RETURN
      END IF
      I=1
      IJ=1
   20 JJ=I+1
      KK=JJ+1
      IF (I.LT.60*IJ) GO TO 40
      WRITE (*,30)
   30 FORMAT (' Hit RETURN to see next page or enter number of phase.')
      IJ=IJ+1
      READ (*,110) BANS
      IF (BANS.EQ.' ') GO TO 40
      READ (BANS,'(i3)',ERR=40) II
      RETURN
   40 IF (KK.EQ.FLIN) GO TO 80
      IF (JJ.EQ.FLIN) GO TO 60
      IF (I.NE.FLIN) GO TO 80
      WRITE (*,50) I,FLINE(I)
   50 FORMAT (I4,': ',A10)
      GO TO 100
   60 WRITE (*,70) I,FLINE(I),JJ,FLINE(JJ)
   70 FORMAT (I4,': ',A10,10X,I3,': ',A10)
      GO TO 100
   80 WRITE (*,90) I,FLINE(I),JJ,FLINE(JJ),KK,FLINE(KK)
   90 FORMAT (I4,': ',A10,10X,I3,': ',A10,10X,I3,': ',A10)
  100 IF (I.EQ.FLIN.OR.I+1.EQ.FLIN.OR.I+2.EQ.FLIN) RETURN
      I=I+3
      GO TO 20
  110 FORMAT (A)
      END
      SUBROUTINE POSCUR (I)
C
C The cursor is positioned at the bottom of a screen of information.
C   I = -1  Cursor is at bottom of main screen
C
$INCLUDE:'NETPATH.BLO'
      WRITE (*,*) CHAR(27)//'[H'
      K=I
      IF (I.EQ.-1) K=ILENGTH+IFLAG(1)+2
      DO 10 J=1,K+4
   10   WRITE (*,*)
      IF (I.EQ.-1) CALL CLPART
      RETURN
      END
      SUBROUTINE PRINT
C
C Results of the mass transfer are displayed.
C
$INCLUDE:'NETPATH.BLO'
      WRITE (IUNIT,10) MODGOOD
   10 FORMAT (//,12X,'MODEL ',I3)
      IF (IFLAG(6).EQ.1) THEN
        IF (IMIX.EQ.1) THEN
```

```
          EVAP=DELTA(3)
          DELTA(1)=DELTA(1)/(1.-EVAP)
          DELTA(2)=DELTA(2)/(1.-EVAP)
        ELSE
          EVAP=DELTA(2)
        END IF
      END IF
      DO 50 I=1,NEQ
        IF (IFLAG(6).EQ.1.AND.IMIX.EQ.1.AND.I.EQ.3) GO TO 50
        IF (IFLAG(6).EQ.1.AND.IMIX.EQ.0.AND.I.LE.2) GO TO 50
        K=NOW(I)
        LL=0
        DO 20 L=1,NOPHA
   20     IF (PHASE(L).EQ.PNAME(K).AND.LL.EQ.0) LL=L
        IBAD=0
        IF (TRANSFER(K).EQ.'-'.AND.DELTA(I).GT.0) IBAD=1
        IF (TRANSFER(K).EQ.'+'.AND.DELTA(I).LT.0) IBAD=1
        IF (IBAD.EQ.0) THEN
          IF (IT(LL).NE.'*') THEN
            WRITE (IUNIT,30) PNAME(K),TRANSFER(K),FORCE(K),DELTA(I)
          ELSE
            WRITE (IUNIT,30) PNAME(K),TRANSFER(K),FORCE(K),DELTA(I),
     *       PARA(LL,1)
          END IF
        ELSE
          WRITE (IUNIT,40) PNAME(K),TRANSFER(K),FORCE(K),DELTA(I)
        END IF
   30   FORMAT (2X,A8,2X,A1,1X,A1,2X,F13.5,:,3X,F10.3,' exchanged')
   40   FORMAT (2X,A8,2X,A1,1X,A1,2X,F13.5,5X,'(Constraint ignored)')
   50 CONTINUE
      IF (IFLAG(6).EQ.1) THEN
        IF (EVAP.GT.0.) THEN
          WRITE (IUNIT,60) 1./(1.-EVAP)
        ELSE
          WRITE (IUNIT,70) (1.-EVAP)
        END IF
      END IF
   60 FORMAT (2X,'Dilution factor: ',F10.3)
   70 FORMAT (2X,'Evaporation factor: ',F10.3)
C     WRITE(IUNIT,50)
      IF (IFLAG(3).EQ.1) CALL CISO (0)
      RETURN
      END
      SUBROUTINE RAYLEIGH (IFILE)
C
C Given initial isotopic values and phase values in and out, the final
C isotopic value is calculated, using the basic m inputs, n outputs
C Rayleigh equation.
C
$INCLUDE:'NETPATH.BLO'
      RESULT=DINIT
      IF (IPRE+IDIS.EQ.0) THEN
        IF (IFILE.EQ.0) WRITE (9,*) 'No phases in or out'
        RETURN
      END IF
      SMGAM=0.
      EPSBAR=0.
      GAMMA=0.
      PHACOMP=0.
      IF (IPRE.EQ.0) PREDAT(1,1)=1.
      DO 10 M=1,IPRE
```

```
        SMGAM=SMGAM+PREDAT(M,1)/PREDAT(1,1)                             BX 190
   10   EPSBAR=EPSBAR+PREDAT(M,1)*PREDAT(M,2)/PREDAT(1,1)               BX 200
C GAMMA AVERAGE ISO COMP. COMPUTATIONS                                  BX 210
      DO 20 N=1,IDIS                                                    BX 220
        PHACOMP=PHACOMP+DISDAT(N,1)/PREDAT(1,1)*DISDAT(N,2)             BX 230
   20   GAMMA=GAMMA+DISDAT(N,1)/PREDAT(1,1)                             BX 240
C CALCULATE EPSBAR/GAMMA                                                BX 250
      IF (GAMMA.EQ.0.) THEN                                             BX 260
        EPSGAM=1000.                                                    BX 270
      ELSE                                                              BX 280
        EPSGAM=EPSBAR/GAMMA                                             BX 290
        PHACOMP=PHACOMP/GAMMA                                           BX 300
      END IF                                                            BX 310
      BETA=1+(.001*EPSGAM)                                              BX 320
      IF (SMGAM-GAMMA.NE.0.) THEN                                       BX 330
        IF (IFILE.EQ.0) WRITE (9,30) 'CFINAL',CFINAL,'CINIT',CINIT      BX 340
        EXPON=(CFINAL/CINIT)**((GAMMA+.001*EPSBAR)/(SMGAM-GAMMA))       BX 350
        IF (IFILE.EQ.0) WRITE (9,30) 'EXPONENT',((GAMMA+.001*EPSBAR)/   BX 360
     *    (SMGAM-GAMMA)),'EPSBAR',EPSBAR                                BX 370
      ELSE                                                              BX 380
        EXPON=EXP(-BETA*GAMMA*PREDAT(1,1)/CINIT)                        BX 390
      END IF                                                            BX 400
      RESULT=((BETA*RESULT-PHACOMP+EPSGAM)*EXPON+PHACOMP-EPSGAM)/BETA   BX 410
      IF (IFILE.EQ.1) RETURN                                            BX 420
      WRITE (9,30) 'BETA',BETA,'PHACOMP',PHACOMP,'EPSGAM',EPSGAM        BX 430
      WRITE (9,30) 'EXPON',EXPON,'SMGAM',SMGAM,'GAMMA',GAMMA            BX 440
   30 FORMAT (3(A,1X,F11.5,1X))                                         BX 450
      RETURN                                                            BX 460
      END                                                               BX 470
      SUBROUTINE RDPATH (FILEONE)                                       BY  10
C                                                                       BY  20
C The data from the .PATH file are read in, including the data that is  BY  30
C passed through untouched by WATEQFP.                                  BY  40
C                                                                       BY  50
$INCLUDE:'NETPATH.BLO'                                                  BY  60
      INTEGER LLINE(8)                                                  BY  70
      CHARACTER*80 FILEONE                                              BY  80
      CHARACTER*32 TEMP                                                 BY  90
      CHARACTER*1 DUMMY(14),STAR(50)                                    BY 100
C THESE DATA NOT AFFECTED BY WATEQFP                                    BY 110
      DATA LLINE/21,22,24,26,27,28,31,32/                               BY 120
      OPEN (UNIT=7,FILE=FILEONE,STATUS='OLD',ERR=10)                    BY 130
      GO TO 20                                                          BY 140
   10 STOP 1                                                            BY 150
   20 REWIND (UNIT=7)                                                   BY 160
      READ (7,30) TEMP                                                  BY 170
   30 FORMAT (///////,A32)                                              BY 180
      NWLLS=0                                                           BY 190
   40 NWLLS=NWLLS+1                                                     BY 200
      I=NWLLS                                                           BY 210
      DO 50 J=1,50                                                      BY 220
   50   STAR(J)=' '                                                     BY 230
      READ (7,60,ERR=110,END=100) TEMP                                  BY 240
      WLLNMS(I)=' '                                                     BY 250
      WLLNMS(I)(5:36)=TEMP                                              BY 260
   60 FORMAT (A32)                                                      BY 270
      READ (7,70,ERR=110) (DBDATA(I,J),STAR(J),J=1,8)                   BY 280
   70 FORMAT (8(F8.4,A1))                                               BY 290
   80 FORMAT (8(F8.4,A1),2X,I5)                                         BY 300
      READ (7,70,ERR=110) (DBDATA(I,J),STAR(J),J=9,16)                  BY 310
      READ (7,70,ERR=110) (DBDATA(I,J),STAR(J),J=17,19),DBDATA(I,25),   BY 320
```

```
     *DUMMY(1),DBDATA(I,29),STAR(29),DBDATA(I,30),STAR(30),CNO3,DUMMY(2) BY 330
     *,CNH4,DUMMY(3)                                                      BY 340
      READ (7,70,ERR=110) DBDATA(I,20),DUMMY(4),(DBDATA(I,J),DUMMY(5),J= BY 350
     *36,39),CC,DUMMY(8),CFE2,DUMMY(7),CFE3,DUMMY(6)                      BY 360
      READ (7,70,ERR=110) CMN2,DUMMY(9),CMN3,DUMMY(10),CMN6,DUMMY(11),    BY 370
     *CMN7,DUMMY(12),CCH4,DUMMY(13),CDOC,DUMMY(14),RSDOC,STAR(48)         BY 380
      DBDATA(I,41)=CC                                                     BY 390
      DBDATA(I,42)=CCH4                                                   BY 400
      DBDATA(I,43)=CDOC                                                   BY 410
      DBDATA(I,44)=-40.                                                   BY 420
      DBDATA(I,45)=-25.                                                   BY 430
      DBDATA(I,46)=0.                                                     BY 440
      DBDATA(I,47)=0.                                                     BY 450
      DBDATA(I,48)=RSDOC                                                  BY 460
      READ (7,80,ERR=110) (DBDATA(I,LLINE(J)),STAR(LLINE(J)),J=1,8),      BY 470
     *TOT(I)                                                              BY 480
      IF (TOT(I).EQ.0) TOT(I)=I                                           BY 490
C Equation for 34SH2S based on 34SSO4:  CAN CHANGE!                       BY 500
C                                                                         BY 510
      IF (STAR(32).EQ.'*'.AND.STAR(31).EQ.' ') THEN                       BY 520
        DBDATA(I,32)=DBDATA(I,31)-54.0+0.4*DBDATA(I,25)                   BY 530
        STAR(32)=' '                                                      BY 540
      END IF                                                              BY 550
      DBDATA(I,20)=DBDATA(I,20)*4.                                        BY 560
      DBDATA(I,24)=DBDATA(I,24)*DBDATA(I,15)                              BY 570
      DBDATA(I,21)=DBDATA(I,21)*DBDATA(I,41)                              BY 580
      DBDATA(I,22)=DBDATA(I,22)*DBDATA(I,41)                              BY 590
      DBDATA(I,31)=DBDATA(I,31)*DBDATA(I,30)                              BY 600
      DBDATA(I,32)=DBDATA(I,32)*DBDATA(I,29)                              BY 610
      DBDATA(I,23)=DBDATA(I,31)+DBDATA(I,32)                              BY 620
      DBDATA(I,33)=CNO3*5-CNH4*3                                          BY 630
C DBDATA( ,34) DOES NOT INCLUDE RS OF DOC, BECAUSE THIS VARIABLE. IT IS   BY 640
C  INCLUDED WHEN THE MODEL IS RUN.                                        BY 650
      DBDATA(I,34)=CC*4-CCH4*4                                            BY 660
      DBDATA(I,35)=CFE2*2+CFE3*3                                          BY 670
      DBDATA(I,40)=CMN2*2+CMN3*3+CMN6*6+CMN7*7                            BY 680
      IF (STAR(31).EQ.'*'.OR.STAR(32).EQ.'*') STAR(23)='*'                BY 690
      IF (STAR(29).EQ.' '.OR.STAR(30).EQ.' ') STAR(2)=' '                 BY 700
      DO 90 ISTAR=1,50                                                    BY 710
        NODATA(I,ISTAR)=0                                                 BY 720
        IF (STAR(ISTAR).EQ.'*') NODATA(I,ISTAR)=1                         BY 730
   90 CONTINUE                                                            BY 740
      GO TO 40                                                            BY 750
  100 CLOSE (7)                                                           BY 760
      NWLLS=NWLLS-1                                                       BY 770
      RETURN                                                              BY 780
  110 WRITE (*,120) NWLLS,J,WLLNMS(NWLLS)(1:79)                           BY 790
  120 FORMAT (/' **** Error ****'/' Unknown error at well#',I3,I3/1X,     BY 800
     *A79/' ***************'/)                                            BY 810
      STOP                                                                BY 820
      END                                                                 BY 830
      SUBROUTINE ROWCOL                                                   BZ  10
C                                                                         BZ  20
C Checks for use of all phases and all constraints in current model.      BZ  30
C                                                                         BZ  40
$INCLUDE:'NETPATH.BLO'                                                    BZ  50
      INTEGER IROW(39),ICOL(39)                                           BZ  60
      ARRAYOK=.TRUE.                                                      BZ  70
      IF ((1+IFLAG(1)+IFLAG(6)).GT.NEQ) ARRAYOK=.FALSE.                   BZ  80
      DO 10 I=1,NEQ                                                       BZ  90
        IROW(I)=0                                                         BZ 100
```

```
        ICOL(I)=0
   10 CONTINUE
      DO 30 I=1,NEQ
        DO 20 J=1,NEQ
          IF (DABS(A(I,J)).GT.1E-10) IROW(I)=1
          IF (DABS(A(J,I)).GT.1E-10) ICOL(I)=1
   20   CONTINUE
        IF (IROW(I).EQ.0) THEN
          ARRAYOK=.FALSE.
        END IF
        IF (ICOL(I).EQ.0) THEN
          ARRAYOK=.FALSE.
        END IF
   30 CONTINUE
      RETURN
      END
      SUBROUTINE RUN (NUMRUN)
C
C The model is run, with output going to the screen or a file.  The
C phases are prepared to be run and all the data is put into the NEWBAL
C format.  NEWBAL is then called.
C
$INCLUDE:'NETPATH.BLO'
      DOUBLE PRECISION PC(15)
      INTEGER JE(15)
      CHARACTER ITTEMP(39)
      NRUN=NUMRUN
      CALL CLS
      IUNIT=6
      IF (NRUN.EQ.0) IUNIT=8
   10 FORMAT (' Initial Well : ',A,/,' Final Well   : ',A,//,'
     *Final     Initial')
   20 FORMAT (' Initial Well 1 : ',A,/,' Initial Well 2 : ',A,/,' Final
     *Well     : ',A//,'         Final    Initial 1   ','Initial 2')
      IF (IFLAG(1).EQ.0) THEN
        WRITE (IUNIT,10) WLLNMS(WELL(1))(5:36),WLLNMS(WELL(3))(5:36)
      ELSE
        WRITE (IUNIT,20) WLLNMS(WELL(1))(5:36),WLLNMS(WELL(2))(5:36),
     *   WLLNMS(WELL(3))(5:36)
      END IF
      DO 50 I=1,NOELE
        ENAME(I)=ELESHORT(IELE(I))
        SFINAL(I)=DBDATA(WELL(3),IELE(I))
        SINIT1(I)=DBDATA(WELL(1),IELE(I))
        IF (IFLAG(1).EQ.1) SINIT2(I)=DBDATA(WELL(2),IELE(I))
        IF (IELE(I).EQ.21) THEN
          SFINAL(I)=SFINAL(I)+DBDATA(WELL(3),42)*DBDATA(WELL(3),44)+
     *     DBDATA(WELL(3),43)*DBDATA(WELL(3),45)
          SINIT1(I)=SINIT1(I)+DBDATA(WELL(1),42)*DBDATA(WELL(1),44)+
     *     DBDATA(WELL(1),43)*DBDATA(WELL(1),45)
          IF (IFLAG(1).EQ.1) SINIT2(I)=SINIT2(I)+DBDATA(WELL(2),42)*
     *     DBDATA(WELL(2),44)+DBDATA(WELL(2),43)*DBDATA(WELL(2),45)
        END IF
        IF (IELE(I).EQ.22) THEN
          SFINAL(I)=SFINAL(I)+DBDATA(WELL(3),42)*DBDATA(WELL(3),46)+
     *     DBDATA(WELL(3),43)*DBDATA(WELL(3),47)
          SINIT1(I)=SINIT1(I)+DBDATA(WELL(1),42)*DBDATA(WELL(1),46)+
     *     DBDATA(WELL(1),43)*DBDATA(WELL(1),47)
          IF (IFLAG(1).EQ.1) SINIT2(I)=SINIT2(I)+DBDATA(WELL(2),42)*
     *     DBDATA(WELL(2),46)+DBDATA(WELL(2),43)*DBDATA(WELL(2),47)
        END IF
```

```
     IF (IELE(I).EQ.20) THEN                                           CA 460
       DO 30 J=1,NOELE                                                 CA 470
         IF (IELE(J).EQ.1) THEN                                        CA 480
           IF (ISDOCRS(3).EQ.0) THEN                                   CA 490
             SFINAL(I)=SFINAL(I)+DBDATA(WELL(3),34)+DBDATA(WELL(3),    CA 500
  *           43)*DBDATA(WELL(3),48)                                   CA 510
           ELSE                                                        CA 520
             SFINAL(I)=SFINAL(I)+DBDATA(WELL(3),34)+DBDATA(WELL(3),    CA 530
  *           43)*DBDATA(WELL(3),49)                                   CA 540
           END IF                                                      CA 550
           IF (ISDOCRS(1).EQ.0) THEN                                   CA 560
             SINIT1(I)=SINIT1(I)+DBDATA(WELL(1),34)+DBDATA(WELL(1),    CA 570
  *           43)*DBDATA(WELL(1),48)                                   CA 580
           ELSE                                                        CA 590
             SINIT1(I)=SINIT1(I)+DBDATA(WELL(1),34)+DBDATA(WELL(1),    CA 600
  *           43)*DBDATA(WELL(1),49)                                   CA 610
           END IF                                                      CA 620
           IF (IFLAG(1).EQ.1) THEN                                     CA 630
             IF (ISDOCRS(2).EQ.0) THEN                                 CA 640
               SINIT2(I)=SINIT2(I)+DBDATA(WELL(2),34)+DBDATA(WELL(2), CA 650
  *             43)*DBDATA(WELL(2),48)                                 CA 660
             ELSE                                                      CA 670
               SINIT2(I)=SINIT2(I)+DBDATA(WELL(2),34)+DBDATA(WELL(2), CA 680
  *             43)*DBDATA(WELL(2),49)                                 CA 690
             END IF                                                    CA 700
           END IF                                                      CA 710
         END IF                                                        CA 720
         IF (IELE(J).EQ.2) THEN                                        CA 730
           SFINAL(I)=SFINAL(I)+DBDATA(WELL(3),30)*6-DBDATA(WELL(3),    CA 740
  *         29)*2                                                      CA 750
           SINIT1(I)=SINIT1(I)+DBDATA(WELL(1),30)*6-DBDATA(WELL(1),    CA 760
  *         29)*2                                                      CA 770
           IF (IFLAG(1).EQ.1) SINIT2(I)=SINIT2(I)+DBDATA(WELL(2),30)* CA 780
  *         6-DBDATA(WELL(2),29)*2                                     CA 790
         END IF                                                        CA 800
         IF (IELE(J).EQ.16) THEN                                       CA 810
           SFINAL(I)=SFINAL(I)+DBDATA(WELL(3),35)                      CA 820
           SINIT1(I)=SINIT1(I)+DBDATA(WELL(1),35)                      CA 830
           IF (IFLAG(1).EQ.1) SINIT2(I)=SINIT2(I)+DBDATA(WELL(2),35)  CA 840
         END IF                                                        CA 850
         IF (IELE(J).EQ.17) THEN                                       CA 860
           SFINAL(I)=SFINAL(I)+DBDATA(WELL(3),40)                      CA 870
           SINIT1(I)=SINIT1(I)+DBDATA(WELL(1),40)                      CA 880
           IF (IFLAG(1).EQ.1) SINIT2(I)=SINIT2(I)+DBDATA(WELL(2),40)  CA 890
         END IF                                                        CA 900
         IF (IELE(J).EQ.18) THEN                                       CA 910
           SFINAL(I)=SFINAL(I)+DBDATA(WELL(3),33)                      CA 920
           SINIT1(I)=SINIT1(I)+DBDATA(WELL(1),33)                      CA 930
           IF (IFLAG(1).EQ.1) SINIT2(I)=SINIT2(I)+DBDATA(WELL(2),33)  CA 940
         END IF                                                        CA 950
30     CONTINUE                                                        CA 960
     END IF                                                            CA 970
     IF (IFLAG(1).EQ.0) SINIT2(I)=0                                    CA 980
     IF (IFLAG(1).EQ.0) THEN                                           CA 990
       WRITE (IUNIT,40) ENAME(I),SFINAL(I),SINIT1(I)                   CA1000
     ELSE                                                              CA1010
       WRITE (IUNIT,40) ENAME(I),SFINAL(I),SINIT1(I),SINIT2(I)         CA1020
     END IF                                                            CA1030
40   FORMAT (1X,A2,3(F12.4))                                           CA1040
50 CONTINUE                                                            CA1050
   IF (NRUN.EQ.0) GO TO 70                                             CA1060
```

```
      WRITE (*,60)                                                      CA1070
   60 FORMAT (' Hit <Q> to quit, or <Enter> to continue')               CA1080
      READ (*,130) ANS                                                  CA1090
      IF (UPCS(ANS).EQ.'Q') GO TO 240                                   CA1100
   70 WRITE (IUNIT,*)                                                   CA1110
      NELTS=NOELE                                                       CA1120
      NEQ=NELTS                                                         CA1130
      IF (IFLAG(1).EQ.1.OR.IFLAG(6).EQ.1) NEQ=NEQ+1                     CA1140
      DO 140 I=1,NOPHA                                                  CA1150
        DO 80 J=1,15                                                    CA1160
          JE(J)=JELE(I,J)                                               CA1170
          PC(J)=PCOEFF(I,J)                                             CA1180
   80   CONTINUE                                                        CA1190
        ITTEMP(I)=IT(I)                                                 CA1200
        IF (IT(I).EQ.'*') ITTEMP(I)=' '                                 CA1210
        IF (UPCS(PHASE(I)).NE.'EXCHANGE') GO TO 90                      CA1220
        JE(1)=3                                                         CA1230
        JE(2)=6                                                         CA1240
        JE(3)=5                                                         CA1250
        JE(4)=0                                                         CA1260
        PC(2)=2.0                                                       CA1270
        IF (IFLAG(2).EQ.1) THEN                                         CA1280
          IF (DBDATA(WELL(3),3)+DBDATA(WELL(3),5).GT.0.) THEN           CA1290
            CA=-DBDATA(WELL(3),3)/(DBDATA(WELL(3),3)+DBDATA(WELL(3),5)) CA1300
          ELSE                                                          CA1310
            CA=-1.                                                      CA1320
          END IF                                                        CA1330
        END IF                                                          CA1340
        IF (IFLAG(2).EQ.2) CA=-0.5                                      CA1350
        IF (IFLAG(2).EQ.3) CA=-1.                                       CA1360
        IF (IFLAG(2).EQ.4) CA=-P(2)                                     CA1370
        PC(1)=CA                                                        CA1380
        PC(3)=-1.-CA                                                    CA1390
   90   IF (UPCS(PHASE(I)).NE.'CO2-CH4 ') GO TO 100                     CA1400
        JE(2)=20                                                        CA1410
        JE(3)=0                                                         CA1420
        PC(2)=(P(1)*8.0)-4.0                                            CA1430
  100   J=0                                                             CA1440
        IQC=0                                                           CA1450
        IQS=0                                                           CA1460
        IQSR=0                                                          CA1470
  110   J=J+1                                                           CA1480
        IF (JE(J).NE.0) GO TO 110                                       CA1490
        J=J-1                                                           CA1500
        CALL INCLISO (JE,PC,I,J,1)                                      CA1510
        PNAME(I)=PHASE(I)                                               CA1520
        TRANSFER(I)=ITTEMP(I)                                           CA1530
        FORCE(I)=F(I)                                                   CA1540
        DO 120 K=1,14                                                   CA1550
          IF (K.LE.J) THEN                                              CA1560
            PELT(I,K)=ELESHORT(JE(K))                                   CA1570
            PCOEF(I,K)=PC(K)                                            CA1580
          ELSE                                                          CA1590
            PELT(I,K)='                                                 CA1600
            PCOEF(I,K)=0                                                CA1610
          END IF                                                        CA1620
  120   CONTINUE                                                        CA1630
  130   FORMAT (A)                                                      CA1640
  140 CONTINUE                                                          CA1650
      NMINS=NOPHA                                                       CA1660
      J=0                                                               CA1670
```

```
      DO 190 I=1,NMINS
        IF (FORCE(I).EQ.'F') THEN
          J=J+1
          PNAME(NMINS+1)=PNAME(I)
          FORCE(NMINS+1)=FORCE(I)
          TRANSFER(NMINS+1)=TRANSFER(I)
          DO 150 K=1,15
            PELT(NMINS+1,K)=PELT(I,K)
  150       PCOEF(NMINS+1,K)=PCOEF(I,K)
          DO 170 LOOP=I,J+1,-1
            PNAME(LOOP)=PNAME(LOOP-1)
            FORCE(LOOP)=FORCE(LOOP-1)
            TRANSFER(LOOP)=TRANSFER(LOOP-1)
            DO 160 K=1,15
              PELT(LOOP,K)=PELT(LOOP-1,K)
  160         PCOEF(LOOP,K)=PCOEF(LOOP-1,K)
  170     CONTINUE
          PNAME(J)=PNAME(NMINS+1)
          FORCE(J)=FORCE(NMINS+1)
          TRANSFER(J)=TRANSFER(NMINS+1)
          DO 180 K=1,15
            PELT(J,K)=PELT(NMINS+1,K)
  180       PCOEF(J,K)=PCOEF(NMINS+1,K)
        END IF
  190 CONTINUE
      DO 210 I=1,NOPHA
        K=0
  200   K=K+1
        IF (PELT(I,K).NE.' ') GO TO 200
        K=K-1
  210   WRITE (IUNIT,220) PNAME(I),(PELT(I,J),PCOEF(I,J),J=1,K)
  220 FORMAT (1X,A8,7(A2,F8.4),:,/,8X,7(A2,F8.4))
      IF (NRUN.EQ.0) GO TO 230
      WRITE (*,60)
      READ (*,130) ANS
      IF (UPCS(ANS).EQ.'Q') GO TO 240
  230 IMIX=IFLAG(1)
      ISOLN=1
      CALL BALN
      CLOSE (8)
  240 CALL SCREEN
      RETURN
      END
      SUBROUTINE SAVE
C
C The user is asked whether the model or results should be saved,
C is prompted for filenames, and the data is saved, if desired.
C
      CHARACTER ANS
      CHARACTER*40 UPCS
      CALL POSCUR (-1)
      WRITE (*,20)
      WRITE (*,10)
   10 FORMAT (' (<Enter> for no)')
   20 FORMAT (' Do you want to save the model?')
      READ (*,50) ANS
      IF (UPCS(ANS).EQ.'Y') CALL SAVEMOD
   30 CALL POSCUR (-1)
      WRITE (*,40)
      WRITE (*,10)
   40 FORMAT (' Do you want to save the results?')
```

```
      READ (*,50) ANS
      IF (UPCS(ANS).EQ.'Y') CALL SAVERUN (*30)
   50 FORMAT (A)
      RETURN
      END
      SUBROUTINE SAVEMOD
C
C The model file, containing ALL the data affecting the model is stored.
C Name duplications are recognized and the user is allowed to change the
C name under which to store the data.
C
$INCLUDE:'NETPATH.BLO'
      INTEGER JE(15)
      DOUBLE PRECISION PC(15)
   10 FORMAT (A)
   20 CALL POSCUR (-1)
      IF (EFILE.EQ.' ') WRITE (*,30)
   30 FORMAT (' Enter file name. <Enter to abort>')
      IF (EFILE.NE.' ') WRITE (*,40) EFILE(1:LENS(EFILE))
   40 FORMAT (' Enter file name.  <Enter> for ',A)
      READ (*,10) LINE
      IF (LINE.EQ.' '.AND.EFILE.EQ.' ') RETURN
      IF (LINE.EQ.' ') GO TO 50
      EFILE=LINE
   50 OPEN (8,FILE=EFILE,STATUS='NEW',ERR=170)
   60 WRITE (8,'(A80)') WFILE
      IF (IFLAG(1).EQ.0) WRITE (8,70) TOT(WELL(1)),TOT(WELL(3))
      IF (IFLAG(1).EQ.1) WRITE (8,70) TOT(WELL(1)),TOT(WELL(2)),
     *TOT(WELL(3))
   70 FORMAT (3I3)
      WRITE (8,80) (ELESHORT(IELE(I)),I=1,NOELE)
   80 FORMAT (26(A3))
      DO 110 I=1,NOPHA
        DO 90 J=1,15
          JE(J)=JELE(I,J)
   90     PC(J)=PCOEFF(I,J)
        J=0
  100   J=J+1
        IF (JE(J).NE.0) GO TO 100
        IF (PARA(I,1).GT.0) THEN
          JE(J)=33
          PC(J)=PARA(I,1)
        ELSE
          J=J-1
        END IF
        CALL INCLISO (JE,PC,I,J,0)
        WRITE (8,120) PHASE(I),F(I),IT(I),(ELESHORT(JE(K)),PC(K),K=1,J)
  110 CONTINUE
  120 FORMAT (A8,2(A1),7(A2,F8.3),:,/,10X,7(A2,F8.3))
      WRITE (8,10)
      IF (IFLAG(1).EQ.0) THEN
        WRITE (8,130) (IFLAG(I),I=2,6),(P(I),I=1,2),(ISDOCRS(I),I=1,3),
     *   DISALONG,(C14DAT(I),I=1,9),I10,I11,(C14DAT(I),I=12,13),
     *   (DBDATA(WELL(3),I),I=44,47),DBDATA(WELL(3),49),(DBDATA(WELL(1),
     *   I),I=44,47),DBDATA(WELL(1),49)
      ELSE
        WRITE (8,130) (IFLAG(I),I=2,6),(P(I),I=1,2),(ISDOCRS(I),I=1,3),
     *   DISALONG,(C14DAT(I),I=1,9),I10,I11,(C14DAT(I),I=12,13),
     *   (DBDATA(WELL(3),I),I=44,47),DBDATA(WELL(3),49),(DBDATA(WELL(1),
     *   I),I=44,47),DBDATA(WELL(1),49),(DBDATA(WELL(2),I),I=44,47),
     *   DBDATA(WELL(2),49)
```

```
      END IF                                                             CC 570
  130 FORMAT (5(I2),2(F8.4),3(I2),F6.3,/,7(F8.3),/,2(F8.3),2(I8),2(F8.3) CC 580
     */,3(5(F8.3),/))                                                    CC 590
      CLOSE (8)                                                          CC 600
      OPEN (UNIT=7,FILE='MODEL.FIL')                                     CC 610
      REWIND (7)                                                         CC 620
  140 READ (7,10,END=150) LINE                                           CC 630
      IF (UPCS(LINE).EQ.UPCS(EFILE)) GO TO 160                           CC 640
      GO TO 140                                                          CC 650
  150 BACKSPACE (7)                                                      CC 660
      WRITE (7,10) EFILE(1:LENS(EFILE))                                  CC 670
  160 CLOSE (7)                                                          CC 680
      RETURN                                                             CC 690
  170 CALL POSCUR (-1)                                                   CC 700
      OPEN (8,FILE=EFILE,ERR=20)                                         CC 710
      WRITE (*,180)                                                      CC 720
  180 FORMAT (' File exists.  Type new filename or <Enter> to',' overwri CC 730
     *te.')                                                              CC 740
      READ (*,10) LINE                                                   CC 750
      IF (LINE.NE.' ') THEN                                              CC 760
        CLOSE (8)                                                        CC 770
        OPEN (8,FILE=LINE,STATUS='NEW',ERR=170)                          CC 780
        EFILE=LINE                                                       CC 790
      END IF                                                             CC 800
      GO TO 60                                                           CC 810
      END                                                                CC 820
      SUBROUTINE SAVEOTHER (IJ)                                          CD  10
C                                                                        CD  20
C After a phase is made from scratch or a current phase is edited, it    CD  30
C may be included in NETPATH.DAT for use during future runs.             CD  40
C                                                                        CD  50
$INCLUDE:'NETPATH.BLO'                                                   CD  60
      INTEGER JE(15)                                                     CD  70
      DOUBLE PRECISION PC(15)                                            CD  80
      CHARACTER*8 MIDDLE                                                 CD  90
      IF (FLIN.EQ.100) RETURN                                            CD 100
      WRITE (*,10) PHASE(IJ)                                             CD 110
   10 FORMAT (/,' Do you want to store ''',A,''' in NETPATH.DAT?'' <Ente CD 120
     *r> for no')                                                        CD 130
      READ (*,20) ANS                                                    CD 140
   20 FORMAT (A)                                                         CD 150
      ANS=UPCS(ANS)                                                      CD 160
      IF (ANS.NE.'Y') RETURN                                             CD 170
      MIDDLE=PHASE(IJ)                                                   CD 180
   30 I=0                                                                CD 190
   40 I=I+1                                                              CD 200
      IF (LGT(MIDDLE,FLINE(I)(1:8)).AND.I.LT.FLIN) GO TO 40              CD 210
      IF (MIDDLE.NE.FLINE(I)(1:8)) GO TO 80                              CD 220
   50 WRITE (*,60)                                                       CD 230
   60 FORMAT (/,' This phase already exists in NETPATH.DAT.',/,' (1) Sav CD 240
     *e phase with different name or (2) Replace phase in',' file',/,' < CD 250
     *Enter> to quit.')                                                  CD 260
      READ (*,20) ANS                                                    CD 270
      IF (ANS.EQ.' ') RETURN                                             CD 280
      READ (ANS,'(i1)',ERR=50) J                                         CD 290
      IF (J.LT.1.OR.J.GT.2) GO TO 50                                     CD 300
      IF (J.EQ.2) GO TO 100                                              CD 310
      WRITE (*,70)                                                       CD 320
   70 FORMAT (/' Enter new phase name.')                                 CD 330
      READ (*,20,ERR=50) MIDDLE                                          CD 340
      IF (MIDDLE.EQ.' ') GO TO 50                                        CD 350
```

```
      GO TO 30
   80 DO 90 J=FLIN,I,-1
   90   FLINE(J+1)=FLINE(J)
      FLIN=FLIN+1
  100 K=0
  110 K=K+1
      IF (JELE(IJ,K).NE.0) GO TO 110
      K=K-1
      DO 120 J=1,15
        JE(J)=JELE(IJ,J)
        PC(J)=PCOEFF(IJ,J)
  120 CONTINUE
      CALL INCLISO (JE,PC,IJ,K,0)
      WRITE (FLINE(I),130) MIDDLE,IT(IJ),(ELESHORT(JE(J)),PC(J),J=1,K)
  130 FORMAT (A8,1X,A1,15(A2,F8.4))
      RETURN
      END
      SUBROUTINE SAVERUN (*)
C
C The results of the run, including isotope data and all C-14 models, if
C applicable, are stored in a file.
C
$INCLUDE:'NETPATH.BLO'
      CHARACTER*80 LINE2
      CALL POSCUR (-1)
      WRITE (*,10)
   10 FORMAT (' Enter file name. (Enter to abort)')
      READ (*,30) LINE
      IF (LINE.EQ.' ') RETURN
   20 OPEN (8,FILE=LINE,STATUS='NEW',ERR=60)
   30 FORMAT (A,A)
   40 CALL RUN (0)
      RETURN
   50 RETURN 1
   60 WRITE (*,70)
   70 FORMAT (' File exists - hit <Enter> to overwrite or enter',' new n
     *ame')
      READ (*,30) LINE2
      IF (LINE2.NE.' ') THEN
        LINE=LINE2
        GO TO 20
      END IF
      OPEN (8,FILE=LINE,ERR=50)
      GO TO 40
      END
      SUBROUTINE SCREEN
C
C The important aspects of the current model are displayed on the
C screen.  The wells, constraints, phases, and key parameters are
C displayed.  Also, warnings about insufficient data or incorrect sets
C of constraints and phases are displayed, when necessary.
C
$INCLUDE:'NETPATH.BLO'
      CHARACTER*80 BLANK
      BLANK=' '
      BLANK(27:27)='|'
      BLANK(56:56)='|'
      IFORCE=0
      IEX=0
      ICO2=0
      DO 10 I=1,NOPHA
```

```
        IF (PHASE(I).EQ.'EXCHANGE') IEX=1
        IF (PHASE(I).EQ.'CO2-CH4 ') ICO2=1
   10   IF (F(I).EQ.'F') IFORCE=IFORCE+1
      IF (IEX.EQ.0) IFLAG(2)=3
      CALL CLS
      WRITE (*,20) WLLNMS(WELL(1))(5:36)
   20 FORMAT (' Initial Well:',A32)
      IF (IFLAG(1).EQ.1) WRITE (*,20) WLLNMS(WELL(2))(5:36)
      IF (IFORCE.EQ.0) THEN
        WRITE (*,30) WLLNMS(WELL(3))(5:36),NOELE,NOPHA
   30   FORMAT (' Final Well  :',A32,/,1X,79('='),/,6X,'Constraints:',
     *   I3,6X,'|',8X,'Phases:',I3,10X,'|',6X,'Parameters'/,1X,26('-'),'
     *|',28('-'),'|',23('-'))
      ELSE
        WRITE (*,40) WLLNMS(WELL(3))(5:36),NOELE,NOPHA
   40   FORMAT (' Final Well  :',A32,/,1X,79('='),/,6X,'Constraints:',
     *   I3,6X,'|',8X,'Phases:',I3,10X,'|',6X,'Parameters'/,1X,26('-'),'
     *|',11('-'),'Forced',11('-'),'|',23('-'))
      END IF
      I=0
   50 I=I+1
      LINE=BLANK
C WRITE CONSTRAINTS
      IF (2*I.LE.NOELE) THEN
        WRITE (LINE,60) (ELELONG(IELE(J)),J=2*I-1,2*I),LINE(26:)
   60   FORMAT (A12,1X,A12,A)
      ELSE IF (2*I-1.EQ.NOELE) THEN
        WRITE (LINE,70) ELELONG(IELE(2*I-1)),LINE(13:)
   70   FORMAT (A12,A)
      END IF
C WRITE PHASES
      IF (3*I-2.LE.IFORCE) THEN
        N=0
        DO 90 M=1,3*I-3
   80     N=N+1
          IF (F(N).NE.'F') GO TO 80
   90   CONTINUE
        L=0
  100   L=L+1
  110   N=N+1
        IF (F(N).NE.'F') GO TO 110
        LINE(19+9*L:27+9*L)=IT(N)//PHASE(N)
        IF (3*I-2+L.LE.IFORCE.AND.L.LT.3) GO TO 100
      END IF
      IF (I.GT.1.AND.IFORCE.GT.3*I-6.AND.IFORCE.LT.3*I-2.AND.IFORCE.LT.
     *NOPHA) LINE(28:55)='----------Unforced----------'
      IF (IFORCE.EQ.0) THEN
        IF (3*I-2.LE.NOPHA) THEN
          N=3*I-3
          L=0
  120     L=L+1
          N=N+1
          LINE(19+9*L:27+9*L)=IT(N)//PHASE(N)
          IF (3*I-2+L.LE.NOPHA.AND.L.LT.3) GO TO 120
        END IF
      ELSE
        L=I-INT((IFORCE+2)/3)-1
        IF (NOPHA-IFORCE.GE.3*L-2.AND.L.GT.0) THEN
          N=0
          DO 140 M=1,3*L-3
  130       N=N+1
```

```
            IF (F(N).EQ.'F') GO TO 130                                  CF 780
  140     CONTINUE                                                      CF 790
          J=0                                                           CF 800
  150     J=J+1                                                         CF 810
  160     N=N+1                                                         CF 820
          IF (F(N).EQ.'F') GO TO 160                                    CF 830
          LINE(19+9*J:27+9*J)=IT(N)//PHASE(N)                           CF 840
          IF (3*L-2+J.LE.NOPHA-IFORCE.AND.J.LT.3) GO TO 150             CF 850
        END IF                                                          CF 860
      END IF                                                            CF 870
C WRITE PARAMETERS                                                      CF 880
      IF (I.EQ.1) THEN                                                  CF 890
        WRITE (LINE,170) LINE(:57),'Mixing: ',YES(IFLAG(1))             CF 900
      END IF                                                            CF 910
      IF (I.EQ.2) THEN                                                  CF 920
        WRITE (LINE,170) LINE(:57),'Evaporation: ',YES(IFLAG(6))        CF 930
      END IF                                                            CF 940
      IF (I.EQ.3) THEN                                                  CF 950
        WRITE (LINE,170) LINE(:57),'Rayleigh Calcs: ',YES(IFLAG(3))     CF 960
      END IF                                                            CF 970
      IF (IEX.EQ.1.AND.I.EQ.4) THEN                                     CF 980
        WRITE (LINE,170) LINE(:57),'Exchange: ',ION(IFLAG(2))           CF 990
      END IF                                                            CF1000
      ISKIP=0                                                           CF1010
      IF (IFLAG(2).EQ.4) ISKIP=1                                        CF1020
      IF (IEX.EQ.1.AND.IFLAG(2).EQ.4.AND.I.EQ.5) THEN                   CF1030
        WRITE (LINE,180) LINE(:57),'X Calcium:     ',P(2)               CF1040
      END IF                                                            CF1050
      IF (ICO2.EQ.1.AND.I.EQ.4+IEX+ISKIP) THEN                          CF1060
        WRITE (LINE,190) LINE(:57),'X CO2 in CO2-CH4:',P(1)             CF1070
      END IF                                                            CF1080
      IF (DBDATA(WELL(1),1).LE.0.0) GO TO 210                           CF1090
      IF (IFLAG(1).EQ.1) THEN                                           CF1100
        IF (DBDATA(WELL(2),1).LE.0.0) GO TO 210                         CF1110
      END IF                                                            CF1120
      IF (IFLAG(1).EQ.0.AND.IFLAG(3).EQ.1.AND.I.EQ.4+IEX+ISKIP+ICO2)    CF1130
     *WRITE (LINE,180) LINE(:57),'Init C-14     ',C14(IFLAG(4),1)       CF1140
      IF (IFLAG(1).EQ.1.AND.IFLAG(3).EQ.1.AND.I.EQ.4+IEX+ISKIP+ICO2)    CF1150
     *WRITE (LINE,200) LINE(:57),'Init 1 C-14  ',C14(IFLAG(4),1)        CF1160
      IF (IFLAG(1).EQ.1.AND.IFLAG(3).EQ.1.AND.I.EQ.4+IEX+ISKIP+ICO2+1)  CF1170
     *WRITE (LINE,200) LINE(:57),'Init 2 C-14  ',C14(IFLAG(4),2)        CF1180
      IF (IFLAG(3).EQ.1.AND.I.EQ.4+IEX+ISKIP+ICO2+1+IFLAG(1))           CF1190
     *WRITE (LINE,170) LINE(:57),'(',MODEL(IFLAG(4))                    CF1200
     *(1:LENS(MODEL(IFLAG(4)))),')'                                     CF1210
  170 FORMAT (A,A,A,A)                                                  CF1220
  180 FORMAT (A,A,F6.2)                                                 CF1230
  190 FORMAT (A,A,F5.2)                                                 CF1240
  200 FORMAT (A,A,F7.2)                                                 CF1250
  210 IF (LINE.NE.BLANK) THEN                                           CF1260
        WRITE (*,'(1X,A79)') LINE                                       CF1270
        GO TO 50                                                        CF1280
      END IF                                                            CF1290
      ILENGTH=I-1                                                       CF1300
      WRITE (*,220)                                                     CF1310
  220 FORMAT (1X,79('='))                                               CF1320
      CALL WARN                                                         CF1330
      RETURN                                                            CF1340
      END                                                               CF1350
      SUBROUTINE SET                                                    CG  10
$INCLUDE:'NETPATH.BLO'                                                  CG  20
C                                                                       CG  30
```

```
C         INITIALIZE THE A ARRAY
C
      DO 10 I=1,39
        DO 10 J=1,39
        A(I,J)=0.0
   10 CONTINUE
C
C              PUT COEFFICIENTS IN THE A ARRAY
C
      DO 30 I=1,NEQ
        K=NOW(I)
        A(I,NEQ+1)=EDELTA(I)
        DO 20 J=1,16
          IF (LELT(K,J).GT.0) THEN
            A(LELT(K,J),I)=PCOEF(K,J)
          END IF
   20   CONTINUE
   30 CONTINUE
      RETURN
      END
      SUBROUTINE SFRACT (FRACTION,ITIME,FRAC1,FRAC2,IERROR)
$INCLUDE:'NETPATH.BLO'
C
C A specific frationation factor is calculated, based on some fraction
C of the initial and final waters.  FRAC1 and FRAC2 are the fractions of
C the two initial wells to use, if applicable.
C Meaning of ITIME 1:  Sulfide-solution at some point along the flowpath
C                  2:  Sulfate-solution at some point along the flowpath
C Meaning of IERROR 0:  No error
C                   1:  Can't compute at final or initial well
C                   2:  Can't compute at final well
C                   3:  Can't compute at initial well
C
      IERROR=0
      IF (NODATA(WELL(3),23).EQ.1.OR.NODATA(WELL(3),2).EQ.1) THEN
        IERROR=2
      END IF
      IF (NODATA(WELL(1),23).EQ.1.OR.NODATA(WELL(1),2).EQ.1) THEN
        IF (IERROR.EQ.2) THEN
          IERROR=1
          RETURN
        END IF
        IERROR=3
      END IF
      IF (IFLAG(1).EQ.1) THEN
        IF (NODATA(WELL(2),23).EQ.1.OR.NODATA(WELL(2),2).EQ.1) THEN
          IF (IERROR.EQ.2) THEN
            IERROR=1
            RETURN
          END IF
          IERROR=3
        END IF
      END IF
      K=1
   10 DION=DBDATA(WELL(K),32)
      AION=DBDATA(WELL(K),29)
      IF (ITIME.EQ.2) THEN
        DION=DBDATA(WELL(K),31)
        AION=DBDATA(WELL(K),30)
      END IF
      DS=DBDATA(WELL(K),23)
```

```
      AS=DBDATA(WELL(K),2)                                               CH 420
      IF (K.EQ.1.AND.IFLAG(1).EQ.1) THEN                                 CH 430
        DION=FRAC1*DION+FRAC2*DBDATA(WELL(2),31)                         CH 440
        IF (ITIME.EQ.3) DION=FRAC1*DION+FRAC2*DBDATA(WELL(2),32)         CH 450
        AION=FRAC1*AION+FRAC2*DBDATA(WELL(2),30)                         CH 460
        IF (ITIME.EQ.3) AION=FRAC1*AION+FRAC2*DBDATA(WELL(2),29)         CH 470
        DS=FRAC1*DS+FRAC2*DBDATA(WELL(2),23)                             CH 480
        AS=FRAC1*AS+FRAC2*DBDATA(WELL(2),2)                              CH 490
      END IF                                                             CH 500
      IF (AION.LE.0.) THEN                                               CH 510
        IF (K.EQ.1.AND.IERROR.EQ.2) IERROR=1                             CH 520
        IF (K.EQ.1.AND.IERROR.EQ.0) IERROR=3                             CH 530
        IF (K.EQ.3.AND.IERROR.EQ.3) IERROR=1                             CH 540
        IF (K.EQ.3.AND.IERROR.EQ.0) IERROR=2                             CH 550
      ELSE                                                               CH 560
        RATIO=DION/AION                                                  CH 570
      END IF                                                             CH 580
      IF (IERROR.EQ.1) RETURN                                            CH 590
      IF (K.EQ.1) THEN                                                   CH 600
        IF (IERROR.NE.3) FRACTION=RATIO-DS/AS                            CH 610
        K=3                                                              CH 620
        GO TO 10                                                         CH 630
      END IF                                                             CH 640
      IF (IERROR.EQ.0) THEN                                              CH 650
        FRACTION=FRACTION*(1.-DISALONG)+DISALONG*(RATIO-DS/AS)           CH 660
      ELSE IF (IERROR.EQ.3) THEN                                         CH 670
        FRACTION=RATIO-DS/AS                                             CH 680
      END IF                                                             CH 690
      RETURN                                                             CH 700
      END                                                                CH 710
      SUBROUTINE SLNQ (N,A,X,SINGULAR)                                   CI  10
      DOUBLE PRECISION A(39,39),X(39),D,B,CC                             CI  20
      INTEGER IOPT(10)                                                   CI  30
C                                                                        CI  40
C     SOLUTION OF N LINEAR EQUATIONS IN N UNKNOWNS BY GAUSSIAN           CI  50
C     ELIMINATION OR DETERMINANT EVALUTAION.                             CI  60
C     A CONTAINS THE MATRIX OF THE COEFFICIENTS AND N INDICATES THE      CI  70
C     ORDER OF THE MATRIX. IF J EQUALS ZERO, D CONTAINS THE VALUE OF THE CI  80
C     DETERMINANT. IF J DOES NOT EQUAL ZERO, X CONTAINS THE N VALUES OF  CI  90
C     THE UNKNOWNS.                                                      CI 100
C                                                                        CI 110
      LOGICAL SINGULAR                                                   CI 120
      SINGULAR=.FALSE.                                                   CI 130
      IOPT(9)=0                                                          CI 140
      J=1                                                                CI 150
      D=0.0                                                              CI 160
      NP1=N+1                                                            CI 170
      NM1=N-1                                                            CI 180
      IF (IOPT(9).NE.1) GO TO 20                                         CI 190
      DO 10 II=1,N                                                       CI 200
        WRITE (8,240) (A(II,JJ),JJ=1,NP1)                                CI 210
   10 CONTINUE                                                           CI 220
   20 IF (N.EQ.1) GO TO 230                                              CI 230
      DO 120 I=1,NM1                                                     CI 240
        M=I                                                              CI 250
        K=I+1                                                            CI 260
        B=A(I,I)                                                         CI 270
        DO 40 L=K,N                                                      CI 280
          IF (DABS(B)-DABS(A(L,I))) 30,40,40                             CI 290
   30     M=L                                                            CI 300
          B=A(L,I)                                                       CI 310
```

```
   40   CONTINUE
        IF (B) 50,160,50
   50   CC=A(M,I)
        IF (I-M) 60,80,60
   60   D=-D
        DO 70 L=I,NP1
          B=A(I,L)
          IF (DABS(CC).LT.1E-30) GO TO 170
          A(I,L)=A(M,L)/CC
   70     A(M,L)=B
        GO TO 100
   80   DO 90 L=I,NP1
          IF (DABS(CC).LT.1E-30) GO TO 170
          A(I,L)=A(I,L)/CC
   90   CONTINUE
  100   D=D*CC
        DO 110 M=K,N
          B=A(M,I)
          DO 110 L=I,NP1
  110     A(M,L)=A(M,L)-B*A(I,L)
  120 CONTINUE
      D=D*A(N,N)
      IF (J) 130,190,130
  130 IF (DABS(A(N,N)).LT.1E-30) GO TO 170
      X(N)=A(N,NP1)/A(N,N)
      K=NM1
  140 M=K+1
      B=0.0
      DO 150 L=M,N
  150   B=B+A(K,L)*X(L)
      X(K)=A(K,NP1)-B
      K=K-1
      IF (K) 160,190,140
  160 IF (J) 170,180,170
  170 IF (IOPT(9).EQ.1) WRITE (8,250)
      SINGULAR=.TRUE.
      RETURN
  180 D=0.0
  190 IF (IOPT(9).NE.1) GO TO 220
      WRITE (8,200)
  200 FORMAT (1X,'DELTA VALUES')
      WRITE (8,210) (X(I),I=1,N)
  210 FORMAT (5X,1P8D10.1)
  220 RETURN
  230 IF (DABS(A(1,1)).LT.1E-30) GO TO 170
      X(1)=A(1,2)/A(1,1)
      GO TO 180
C
C
  240 FORMAT (/(5X,1P10D12.3))
  250 FORMAT (' MINERAL COMPOSITIONS ARE  NOT LINEARLY INDEPENDENT.')
      END
      SUBROUTINE TRANS (IJ)
C
C The user is asked whether the direction of transfer of a phase should
C be limited.  Also, a phase can be forced to be included in all models
```

```
C considered.
C
$INCLUDE:'NETPATH.BLO'
   10 FORMAT (A)
   20 IF (IT(IJ).EQ.'+') WRITE (*,30) PHASE(IJ),'dissolution'
      IF (IT(IJ).EQ.'-') WRITE (*,30) PHASE(IJ),'precipitation'
      IF (IT(IJ).EQ.'*') WRITE (*,30) PHASE(IJ),'isotopic exchange'
      IF (IT(IJ).EQ.' ') WRITE (*,30) PHASE(IJ),'both'
   30 FORMAT (/,' Enter transfer allowed for ''',A,'''. (1)',' dissoluti
     *on, (2) precipitation,',/,' (3) both, or (4) isotopic',' exchange.
     *',/,' Hit <Enter> for ',A,'.')
      READ (*,10) LINE
      IF (LINE.EQ.' ') THEN
        IF (IT(IJ).EQ.'*') GO TO 40
        GO TO 80
      END IF
      READ (LINE,'(i1)',ERR=20) ITR
      IF (ITR.LT.1.OR.ITR.GT.4) GO TO 20
      IF (ITR.EQ.1) IT(IJ)='+'
      IF (ITR.EQ.2) IT(IJ)='-'
      IF (ITR.EQ.3) IT(IJ)=' '
      IF (ITR.NE.4) THEN
        PARA(IJ,1)=0.0
        GO TO 80
      END IF
   40 IT(IJ)='*'
   50 WRITE (*,60)
      IF (PARA(IJ,1).GT.0.0) WRITE (*,70) PARA(IJ,1)
   60 FORMAT (/,' Enter amount to exchange.  (0 to cancel).')
   70 FORMAT (' <Enter> for ',F9.3)
      READ (*,10) LINE
      IF (LINE.EQ.' '.AND.PARA(IJ,1).GT.0.0) GO TO 80
      READ (LINE,'(f10.0)',ERR=50) PNUM
      IF (PNUM.LE.0.0) GO TO 20
      PARA(IJ,1)=PNUM
   80 IF (F(IJ).EQ.' ') WRITE (*,90) PHASE(IJ),'no.'
      IF (F(IJ).EQ.'F') WRITE (*,90) PHASE(IJ),'yes.'
   90 FORMAT (/,' Should ''',A,''' be included in every model?',/,' <Ent
     *er> for ',A)
      READ (*,10) ANS
      ANS=UPCS(ANS)
      IF (ANS.EQ.'Y') F(IJ)='F'
      IF (ANS.EQ.'N') F(IJ)=' '
      RETURN
      END
      CHARACTER*40 FUNCTION UPCS(LINE)
C
C All the letters in a line are changed to upper case.
C
      CHARACTER*(*) LINE
      UPCS=LINE
      DO 10 I=1,LEN(LINE)
        ICH=ICHAR(LINE(I:I))
   10   IF (ICH.GE.ICHAR('a').AND.ICH.LE.ICHAR('z')) UPCS(I:I)=CHAR(ICH-
     *    ICHAR('a')+ICHAR('A'))
      RETURN
      END
      SUBROUTINE WARN
```

```
C
C This subroutine prints warnings if data is missing or if the model
C will not run.
C Called by SCREEN every time the screen is drawn.
C
$INCLUDE:'NETPATH.BLO'
      DIMENSION INDATA(3)
      NUMWARN=0
C
C Check to see if there are enough constraints (EVAP and Mixing each tak
C up one constraint).
C
      IF (NOELE-IFLAG(1)-IFLAG(6).GT.NOPHA) THEN
        WRITE (*,10) NOELE-IFLAG(1)-IFLAG(6)-NOPHA
        NUMWARN=NUMWARN+1
      END IF
   10 FORMAT (' There are ',I2,' too few phases for the given const','ra
     *ints. No models will run.')
C
C Check to see if there are any constraints that aren't in any of the
C phases.
C
      NBADCON=0
      IFIRST=0
      ISECOND=0
      IC=0
      IS=0
      ISR=0
      DO 50 ICOUNT=1,NOELE
        IF (IELE(ICOUNT).EQ.1) IC=1
        IF (IELE(ICOUNT).EQ.2) IS=1
        IF (IELE(ICOUNT).EQ.15) ISR=1
        ICHECK=IELE(ICOUNT)
C For isotopes, check for element, not isotope
        IF (ICHECK.EQ.21.OR.ICHECK.EQ.22) ICHECK=1
        IF (ICHECK.EQ.23) ICHECK=2
        IF (ICHECK.EQ.24) ICHECK=15
        J=0
   20   J=J+1
C If there are no more phases, this constraint is bad.
        IF (J.GT.NOPHA) GO TO 40
        K=0
   30   K=K+1
C If there aren't more constraints in this phase, check the next one
        IF (JELE(J,K).EQ.0) GO TO 20
C If the constraint found, go to the next constraint to check
        IF (JELE(J,K).EQ.ICHECK) GO TO 50
C Handle special cases
        IF (PHASE(J).EQ.'EXCHANGE'.AND.ICHECK.EQ.3.OR.ICHECK.EQ.5.OR.
     *   ICHECK.EQ.6) GO TO 50
        IF (PHASE(J).EQ.'CO2-CH4 '.AND.ICHECK.EQ.20) GO TO 50
        GO TO 30
C Oh no! The element isn't in any of the phases
   40   NBADCON=NBADCON+1
        IF ((IFLAG(1)+IFLAG(6)).EQ.0) WRITE (*,60) ELELONG(IELE(ICOUNT))
     *    (1:LENS(ELELONG(IELE(ICOUNT))))
        IF ((IFLAG(1)+IFLAG(6)).EQ.1) THEN
          IF (IFIRST.NE.0) THEN
            IF (NBADCON.EQ.2) WRITE (*,60) ELELONG(IELE(IFIRST))
     *        (1:LENS(ELELONG(IELE(IFIRST))))
            WRITE (*,60) ELELONG(IELE(ICOUNT))
```

```
     *       (1:LENS(ELELONG(IELE(ICOUNT))))
          ELSE
            IFIRST=ICOUNT
          END IF
        END IF
        IF ((IFLAG(1)+IFLAG(6)).EQ.2) THEN
          IF (IFIRST.GT.0.AND.ISECOND.GT.0) THEN
            IF (NBADCON.EQ.3) THEN
              WRITE (*,60) ELELONG(IELE(IFIRST))
     *          (1:LENS(ELELONG(IELE(IFIRST))))
              WRITE (*,60) ELELONG(IELE(ISECOND))
     *          (1:LENS(ELELONG(IELE(ISECOND))))
            END IF
            WRITE (*,60) ELELONG(IELE(ICOUNT))
     *        (1:LENS(ELELONG(IELE(ICOUNT))))
          END IF
          IF (IFIRST.GT.0.AND.ISECOND.EQ.0) ISECOND=ICOUNT
          IF (IFIRST.EQ.0) IFIRST=ICOUNT
        END IF
   50 CONTINUE
   60 FORMAT (1X,A,' is not contained in any phase. No models will run.'
     *)
C If only one constraint wasn't included in a phase, and mixing or evap
C are on, we have to tell the user that this one constraint will determi
C the mixing ration or evaporation/dilution factor
      IF (IFLAG(1)+IFLAG(6).EQ.1.AND.NBADCON.EQ.1) THEN
        IF (IFLAG(1).EQ.1) THEN
          WRITE (*,70) ELELONG(IELE(IFIRST))
        ELSE
          WRITE (*,80) ELELONG(IELE(IFIRST))
        END IF
      END IF
   70 FORMAT (' The mixing ratio will be determined by ',A)
   80 FORMAT (' The evaporation/dilution factor will be determined by ',
     *A)
C No warnings were printed if evap and mixing are both on and 2 or fewer
C  constraints don't have corresponding phases.  We have to increase
C  NUMWARN in all other cases.
      IF ((IFLAG(1)+IFLAG(6)).LT.2.OR.NBADCON.GT.2) NUMWARN=NUMWARN+
     *NBADCON
C
C Now we check for phases that won't do anything (they have none of the
C  chosen constraints.
C
      NBADPHA=0
      DO 130 ICOUNT=1,NOPHA
        J=0
   90   J=J+1
        IF (JELE(ICOUNT,J).EQ.0.AND.PHASE(ICOUNT).NE.'EXCHANGE') GO TO
     *   110
        K=0
  100   K=K+1
        IF (K.GT.NOELE) THEN
          IF (PHASE(ICOUNT).EQ.'EXCHANGE') THEN
            GO TO 110
          ELSE
            GO TO 90
          END IF
        END IF
        IF (JELE(ICOUNT,J).EQ.IELE(K)) GO TO 130
        IF (PHASE(ICOUNT).EQ.'EXCHANGE') THEN
```

```
          IF (IELE(K).EQ.3.OR.IELE(K).EQ.5.OR.IELE(K).EQ.6) GO TO 130   CL1240
        END IF                                                          CL1250
        IF (PHASE(ICOUNT).EQ.'CO2-CH4') THEN                            CL1260
          IF (IELE(K).EQ.20) GO TO 130                                  CL1270
        END IF                                                          CL1280
        GO TO 100                                                       CL1290
  110   NBADPHA=NBADPHA+1                                               CL1300
        WRITE (*,120) PHASE(ICOUNT)(1:LENS(PHASE(ICOUNT)))              CL1310
  120   FORMAT (1X,A,' does not contain any of the chosen constraints.') CL1320
  130 CONTINUE                                                          CL1330
      NUMWARN=NUMWARN+NBADPHA                                           CL1340
C                                                                       CL1350
C Phase check is done, now it's time for check whether data for constrai CL1360
C was present in .PAT file                                              CL1370
C                                                                       CL1380
      DO 150 ICOUNT=1,NOELE                                             CL1390
        INDATA(1)=0                                                     CL1400
        INDATA(2)=0                                                     CL1410
        INDATA(3)=0                                                     CL1420
        ICHECK=IELE(ICOUNT)                                             CL1430
        IF (NODATA(WELL(1),ICHECK).EQ.1) INDATA(1)=1                    CL1440
        IF (IFLAG(1).EQ.1) THEN                                         CL1450
          IF (NODATA(WELL(2),ICHECK).EQ.1) INDATA(2)=1                  CL1460
        END IF                                                          CL1470
        IF (NODATA(WELL(3),ICHECK).EQ.1) INDATA(3)=1                    CL1480
        ITOTAL=INDATA(1)+INDATA(2)+INDATA(3)                            CL1490
        IF (ITOTAL.EQ.0) GO TO 150                                      CL1500
        NUMWARN=NUMWARN+1                                               CL1510
        WRITE (*,140) ELELONG(ICHECK)(1:LENS(ELELONG(ICHECK))),ITOTAL   CL1520
  140   FORMAT (' Warning: There is no data for ',A,' in ',I1,' of the', CL1530
     *    ' wells - zero will be used.')                                CL1540
  150 CONTINUE                                                          CL1550
C Print the bottom line of ='s and get out...                           CL1560
C                                                                       CL1570
      IF (NUMWARN.GT.0) THEN                                            CL1580
        ILENGTH=ILENGTH+NUMWARN+1                                       CL1590
        WRITE (*,160)                                                   CL1600
      END IF                                                            CL1610
  160 FORMAT (1X,79('='))                                               CL1620
      RETURN                                                            CL1630
      END                                                               CL1640
      SUBROUTINE WELLFILE                                               CM  10
C                                                                       CM  20
C The initial well data file to be used is selected here.  The program  CM  30
C terminates if no well files have been prepared.                       CM  40
C                                                                       CM  50
$INCLUDE:'NETPATH.BLO'                                                  CM  60
      CHARACTER*80 FILES(100),OFILES(100),FILEONE                       CM  70
      NFILES=0                                                          CM  80
      OPEN (UNIT=7,FILE='NETPATH.FIL',STATUS='OLD',ERR=40)              CM  90
      REWIND (7)                                                        CM 100
   10 READ (7,150,ERR=10,END=40) LINE                                   CM 110
      DO 20 I=34,1,-1                                                   CM 120
        IF (LINE(I:I).NE.' ') GO TO 30                                  CM 130
   20 CONTINUE                                                          CM 140
      GO TO 10                                                          CM 150
   30 LINE(I+1:)='.PAT'                                                 CM 160
      OPEN (8,FILE=LINE,STATUS='OLD',ERR=10)                            CM 170
      CLOSE (8)                                                         CM 180
      NFILES=NFILES+1                                                   CM 190
      FILES(NFILES)=UPCS(LINE(1:I))                                     CM 200
```

```
      GO TO 10                                                          CM 210
   40 IF (NFILES.GT.0) GO TO 50                                         CM 220
   50 CLOSE (7)                                                         CM 230
      CLOSE (8)                                                         CM 240
      IJ=1                                                              CM 250
   60 ICOUNT=(IJ-1)*15                                                  CM 260
      CALL CLS                                                          CM 270
      WRITE (*,70)                                                      CM 280
   70 FORMAT (' ----------------',/,' CHOOSE WELL FILE',/,' ------------ CM 290
     *----',/)                                                          CM 300
   80 ICOUNT=ICOUNT+1                                                   CM 310
   90 FORMAT (I4,':  ',A40)                                             CM 320
      WRITE (*,90) ICOUNT,FILES(ICOUNT)                                 CM 330
      IF (ICOUNT.LT.NFILES.AND.ICOUNT.LT.IJ*15) GO TO 80                CM 340
      DO 100 I=ICOUNT,(IJ*15-1)                                         CM 350
        WRITE (*,*)                                                     CM 360
  100 CONTINUE                                                          CM 370
      IF (IJ*15.LT.NFILES) THEN                                         CM 380
        IJ=IJ+1                                                         CM 390
      ELSE                                                              CM 400
        IJ=1                                                            CM 410
      END IF                                                            CM 420
      IF (NFILES.LE.15.AND.IEDIT.NE.2) WRITE (*,110)                    CM 430
      IF (NFILES.GT.15.AND.IEDIT.NE.2) WRITE (*,120)                    CM 440
      IF (NFILES.LE.15.AND.IEDIT.EQ.2) WRITE (*,130) WFILE              CM 450
      IF (NFILES.GT.15.AND.IEDIT.EQ.2) WRITE (*,140) WFILE              CM 460
  110 FORMAT (/,' Enter number of file to use:')                        CM 470
  120 FORMAT (/,' Enter number of file to use or <ENTER> to see more',' CM 480
     *files:')                                                          CM 490
  130 FORMAT (/,' Enter number of file to use or <ENTER> to keep ',A)   CM 500
  140 FORMAT (/,' Enter number of file to use, ''M'' to see more choices CM 510
     *',', or <ENTER> to keep ',A)                                      CM 520
      READ (*,150) LINE                                                 CM 530
  150 FORMAT (A80)                                                      CM 540
      IF (LINE.EQ.' '.AND.IEDIT.EQ.2) RETURN                            CM 550
      IF (UPCS(LINE).EQ.'M'.OR.LINE.EQ.' ') GO TO 60                    CM 560
      READ (LINE,'(I3)',ERR=60) I                                       CM 570
      IF (I.LT.1.OR.I.GT.NFILES) GO TO 50                               CM 580
      FILEONE=FILES(I)                                                  CM 590
      DO 160 J=74,1,-1                                                  CM 600
        IF (FILEONE(J:J).NE.' ') GO TO 170                              CM 610
  160 CONTINUE                                                          CM 620
      GO TO 50                                                          CM 630
  170 WFILE=FILES(I)                                                    CM 640
      FILEONE(J+1:)='.PAT'                                              CM 650
      OPEN (UNIT=7,FILE='NETPATH.FIL',STATUS='OLD')                     CM 660
      CLOSE (7,STATUS='DELETE')                                         CM 670
      OPEN (UNIT=7,FILE='NETPATH.FIL',STATUS='NEW')                     CM 680
      DO 180 I=1,NFILES                                                 CM 690
        WRITE (7,'(A)') FILES(I)(1:LENS(FILES(I)))                      CM 700
  180 CONTINUE                                                          CM 710
      CLOSE (7)                                                         CM 720
      CALL RDPATH (FILEONE)                                             CM 730
      IF (IEDIT.EQ.2) THEN                                              CM 740
C Note, the next line resets IEDIT to 0, so MODELS works properly.      CM 750
        CALL INITVALS (1)                                               CM 760
        CALL MODELS                                                     CM 770
      END IF                                                            CM 780
      RETURN                                                            CM 790
      END                                                               CM 800
      SUBROUTINE WELLS                                                  CN  10
```

```
C                                                                       CN  20
C The specific wells to be used are selected here, as well as whether   CN  30
C mixing will be considered.                                            CN  40
C                                                                       CN  50
$INCLUDE:'NETPATH.BLO'                                                  CN  60
      CALL CLS                                                          CN  70
      IF (IEDIT.EQ.3) IEDIT=0                                           CN  80
      IF (IEDIT.EQ.0) THEN                                              CN  90
        IFLAG(1)=0                                                      CN 100
        WELL(1)=0                                                       CN 110
        WELL(2)=0                                                       CN 120
        WELL(3)=0                                                       CN 130
      END IF                                                            CN 140
      CALL EDITMIX (1)                                                  CN 150
   10 WRITE (*,20) 1                                                    CN 160
   20 FORMAT (21X,'Initial Well',I2,/)                                  CN 170
      CALL WLLIST (1)                                                   CN 180
      IF (WELL(1).LT.1.OR.WELL(1).GT.NWLLS) GO TO 10                    CN 190
      IF (IFLAG(1).EQ.0) GO TO 40                                       CN 200
   30 WRITE (*,20) 2                                                    CN 210
      CALL WLLIST (2)                                                   CN 220
      IF (WELL(2).LT.1.OR.WELL(2).GT.NWLLS) GO TO 30                    CN 230
   40 WRITE (*,50)                                                      CN 240
   50 FORMAT (21X,'Final Well',/)                                       CN 250
      CALL WLLIST (3)                                                   CN 260
      IF (WELL(3).LT.1.OR.WELL(3).GT.NWLLS) GO TO 40                    CN 270
      WRITE (*,60) WLLNMS(WELL(1))(5:36)                                CN 280
      IF (IFLAG(1).EQ.1) WRITE (*,70) WLLNMS(WELL(2))(5:36)             CN 290
      WRITE (*,80) WLLNMS(WELL(3))(5:36)                                CN 300
   60 FORMAT (///,' Initial well: ',A32)                                 CN 310
   70 FORMAT (' Initial well: ',A32)                                    CN 320
   80 FORMAT (' Final well  : ',A32,//' Are these correct?',' (<Enter> f CN 330
     *or yes)')                                                         CN 340
      IF (.NOT.YN('Y')) GO TO 10                                        CN 350
      RETURN                                                            CN 360
      END                                                               CN 370
      SUBROUTINE WLLIST (II)                                            CO  10
C                                                                       CO  20
C The wells in a given well file are listed.  After 40 wells, the number CO  30
C of the well to be used may be entered, instead of having to see the   CO  40
C rest of the wells.                                                    CO  50
C                                                                       CO  60
$INCLUDE:'NETPATH.BLO'                                                  CO  70
   10 I=1                                                               CO  80
      IJ=1                                                              CO  90
   20 JJ=I+1                                                            CO 100
      IF (I.LT.40*IJ) GO TO 60                                          CO 110
      WRITE (*,30)                                                      CO 120
   30 FORMAT (' Hit RETURN to see next page or enter number of well',' t CO 130
     *o use.')                                                          CO 140
      IJ=IJ+1                                                           CO 150
      READ (*,40) BANS                                                  CO 160
   40 FORMAT (A)                                                        CO 170
      IF (BANS.EQ.'  ') GO TO 60                                        CO 180
      READ (BANS,50,ERR=60) WELL(II)                                    CO 190
      ISDOCRS(II)=0                                                     CO 200
   50 FORMAT (I3)                                                       CO 210
      RETURN                                                            CO 220
   60 IF (I.NE.NWLLS) GO TO 80                                          CO 230
      WRITE (*,70) I,WLLNMS(I)(5:36)                                    CO 240
   70 FORMAT (I4,': ',A32)                                              CO 250
```

```
      GO TO 100
C
   80 WRITE (*,90) I,WLLNMS(I)(5:36),JJ,WLLNMS(JJ)(5:36)
   90 FORMAT (I4,': ',A32,5X,I3,': ',A32)
      GO TO 100
C
  100 IF (I.EQ.NWLLS.OR.I+1.EQ.NWLLS) GO TO 110
      I=I+2
      GO TO 20
  110 WRITE (*,120)
      IF (IEDIT.GT.0.AND.WELL(II).GT.0) WRITE (*,130) WLLNMS(WELL(II))
     *(5:36)
  120 FORMAT (/,' Input number of well.')
  130 FORMAT (' <Enter> for ',A32)
      READ (*,'(a)') LINE
      IF (LINE.EQ.' '.AND.IEDIT.EQ.0) GO TO 10
      IF (LINE.EQ.' ') RETURN
      READ (LINE,50,ERR=110) IIJ
      IF (IIJ.LT.1.OR.IIJ.GT.NWLLS) GO TO 110
      WELL(II)=IIJ
      ISDOCRS(II)=0
      RETURN
      END
      LOGICAL FUNCTION YN(DEF)
C
C A logical true/false is returned as an answer to a yes/no question.
C
      CHARACTER*1 YESNO,DEF
C
   10 READ (*,20) YESNO
   20 FORMAT (A1)
      YN=.FALSE.
      IF (YESNO.EQ.' ') YESNO=DEF
      IF (YESNO.EQ.'Y'.OR.YESNO.EQ.'y') GO TO 40
      IF (YESNO.EQ.'N'.OR.YESNO.EQ.'n') RETURN
      WRITE (*,30)
   30 FORMAT (' Please ''Y'' or ''N''.')
      GO TO 10
   40 YN=.TRUE.
      RETURN
      END
```

ATTACHMENT D:

Listing of NETPATH.BLO and Definition of Key Variables in NETPATH

```
C Some of the Variables in NETPATH.BLO
C
C PNAME(39), PHASE(39)           : The names of the selected phases
C WLLNMS(0:50)                   : The names of the wells
C EFILE,PFILE,WFILE              : Files used for input or output
C ENAME(39),PELT(39,39)          : Short names of constraints
C ELESHORT(33),ELELONG(0:28)     : The lists of short and long names of
C                                  the constraints
C FLINE(100)                     : Lines in NETPATH.DAT - stored phases
C TRANSFER(39),IT(39)            : Transfer limitations on the phases
C FORCE(39),F(39)                : Forcing of phases
C WELL(3)                        : The well numbers for 1 - Initial Well
C                                  2 - Init well 2 (mixing) 3 - Final
C TOT(50)                        : Permanent well numbers (not affected
C                                  by deletions or moves in DB)
C PCOEFF(39,15)                  : Coefficients of elements in phases
C JELE(39,15)                    : short names of elements in phases
C IFLAG(6)                       : 6 flags:
C                                  1 - Mixing  2 - Ion Exchange
C                                  3 - Rayleigh calcs  4 - AO model
C                                  5 - Mook/Deines fract. factors
C                                  6 - Evaporation/Dilution
C P(3)                           : 3 parameters:
C                                  1 - fraction CO2 in CO2-CH4
C                                  2 - fraction calcium in exchange
C                                  3 - <not used>
C IELE(36)                       : The chosen constraints
C DBDATA(50,50)                  : The well data, as follows:
C    1: total Carbon               26: Deuterium
C    2: total Sulfur               27: Oxygen-18
C    3: total Calcium              28: Tritium
C    4: total Aluminum             29: total Sulfides
C    5: total Magnesium            30: total Sulfate
C    6: total Sodium               31: S-34 of SO4 * Sulfate
C    7: total Potassium            32: S-34 of H2S * Sulfides
C    8: total Chloride             33: Nitrogen part of RS
C    9: total Flouride             34: Carbon part of RS (except DOC)
C   10: total Silica               35: Iron part of RS
C   11: total Bromide              36: HCO3
C   12: total Boron                37: pH
C   13: total Barium               38: H2CO3
C   14: total Lithium              39: CO3
C   15: total Strontium            40: Manganese part of RS
C   16: total Iron                 41: total DIC
C   17: total Manganese            42: total CH4
C   18: total Nitrogen             43: total DOC
C   19: total Phosphorous          44: C-13 of CH4
C   20: Dissolved oxygen           45: C-13 of DOC
C   21: C-13 of TDIC * TDIC        46: C-14 of CH4
C   22: C-14 of TDIC * TDIC        47: C-14 of DOC
C   23: S-34 of water * total S    48: RS of DOC
C   24: Sr-87 of water * total Sr 49: User-entered RS of DOC
C   25: Temperature                50: <not used>
C PARA(39,10)                    : Isotopic values for each phase
C NODATA(50,50)                  : Stores whether each value has
```

```
C                                 been entered in DB
C C14DAT(13)                     : Parameters for the A0 models:
C        1: C-14 activity in carbonate minerals
C        2: C-14 activity in soil gas CO2
C        3: C-13 activity in solution
C        4: C-13 of carbonate minerals
C        5: C-13 of soil gas CO2
C        6: User-defined A0 for initial well 1
C        7: User-defined A0 for initial well 2
C        8: C-14 activity in dolomite
C        9: C-14 activity in calcite
C       10: Choice of method for C-13 of soil gas CO2
C       11: Choice of method for C-13 of initial solution
C       12: C-13 of dolomite
C       13: C-13 of calcite
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*8 PNAME(39),PHASE(39)
      CHARACTER*80 LINE,WLLNMS(0:50),EFILE,PFILE,WFILE
      CHARACTER*12 ELELONG(0:28)
      CHARACTER*2 ENAME(39), PELT(39,39),ELESHORT(33)
      CHARACTER*256 FLINE(100)
      CHARACTER*1 TRANSFER(39), FORCE(39),ANS,IT(39),F(39)
      CHARACTER ESC
      CHARACTER*20 MODEL(9)
      CHARACTER*40 UPCS
      CHARACTER*3 YES(0:1),BANS
      CHARACTER*10 ION(4)
      CHARACTER*14 FFACT(0:1)
      LOGICAL YN
      INTEGER TUNIT,RUNIT,WUNIT,RWUNIT,WELL(3),FLIN,FINDTOT
     *,TOT(50)
      DIMENSION PCOEFF(39,15),JELE(39,15),IFLAG(6),P(3),IELE(36)
     C,DBDATA(0:50,0:50),PARA(39,10),NODATA(50,50),C14DAT(13)
      DIMENSION PREDAT(39,2),DISDAT(39,2),ISDOCRS(3)
      DIMENSION SFINAL(39),SINIT1(39),SINIT2(39),EDELTA(39),PCOEF(39,39)
     *,A(39,39),DELTA(39),MAXPOS(39), MINPOS(39), NOW(39), LELT(39,39)
      LOGICAL FIRST, QUIT, ARRAYOK, MODELOK, SINGULAR
      COMMON /CHAR1/ ENAME, PELT, PNAME, TRANSFER, FORCE,PHASE,IT,F
      COMMON /CHAR2/ ESC,YES,ION,MODEL,WLLNMS,FFACT
      COMMON /CHAR3/ EFILE,PFILE,WFILE,ELELONG,ELESHORT,FLINE
      COMMON /LOGIC1/ FIRST, QUIT, ARRAYOK, MODELOK, SINGULAR
      COMMON /DP1/ SFINAL, SINIT1, SINIT2,CINIT,DINIT,CFINAL,DFINAL
      COMMON /DP2/ PREDAT,DISDAT,RESULT, EDELTA, PCOEF, A, DELTA
      COMMON /DP3/ DISALONG,PCOEFF,PARA,DBDATA,P,C14DAT
      COMMON /INT1/ MAXPOS, MINPOS, NOW, LELT,IPRE,IDIS,NWLLS
      COMMON /INT2/ICHANFORM,IADD,IQUIT,IEDIT,IOLD,TUNIT,RUNIT,WUNIT
      COMMON /INT3/IZEROS,NERR,NRUN,IFLA,RWUNIT,IUNIT,ILENGTH,NOPHA
      COMMON /INT4/  NEQ, NMINS, MODTOT, MODGOOD, IMIX, NELTS, ISOLN
      COMMON /INT5/ JELE, WELL,NODATA,TOT, IFLAG, FLIN, IELE,NOELE
      COMMON /INT6/ ISDOCRS
      EQUIVALENCE (C14DAT(10),I10)
      EQUIVALENCE (C14DAT(11),I11)
```

ATTACHMENT E:

Listing of the File NETPATH.DAT

```
"CH2O"   +C  1.000  I1-25.00  I2  0.00
ALBITE   +NA 1.000  AL 1.000  SI 3.000
ALUNITE   K  1.000  AL 3.000  S  2.000  RS 12.000
ANALCIME +NA 1.000  AL 1.000  SI 2.000
ANNITE   +K  1.000  FE 3.000  AL 1.000  SI 3.000  RS 6.000
ANORTH   +CA 1.000  AL 2.000  SI 2.000
ARAGONIT  CA  1.0000C   1.0000RS  4.0000
BARITE    BA 1.000  S  1.000  RS 6.000
BIOTITE   AL  1.0000MG  3.0000K   1.0000SI  3.0000
BRUCITE   MG 1.000
Biotite   AL  1.0000MG  3.0000K   1.0000SI  3.0000
CALCITE   CA 1.000  C  1.000  RS 4.000  I1 0.000  I2 0.000
CELESTIT +SR 1.000  S  1.000  RS 6.000
CH4 GAS   C  1.000  RS-4.000  I1-40.00  I2 0.000
CHLORITE +MG 5.000  AL 2.000  SI 3.000
CHRYSOTL +MG 3.000  SI 2.000
CO2 GAS   C   1.0000RS  4.0000I1-25.0000I2100.0000
CO2-CH4   C   1.0000I1-25.0000I2100.0000I3-40.00  I4 0.000
Ca-Al-PX  CA 1.000  AL 2.000  SI 1.000
Ca-MONT   CA 0.167  AL 2.330  SI 3.670
DIOPSIDE +CA 1.000  MG 1.000  SI 2.000
DOLOMITE +CA 1.000  MG 1.000  C  2.000  RS 8.000  I1 0.000  I2 0.000
EXCHANGE
FLUORAP  +CA 5.000  P  4.000  F  1.000
FLUORITE +CA 1.000  F  2.000
FORSTRIT +MG 2.000  SI 1.000
Fe-S     -FE 1.000  S  1.000  RS 0.000
FeII-Na   FE  1.0000NA -2.0000RS  2.0000
GIBBSITE  AL 1.000
GOETHITE  FE 1.000  RS 3.000
GYPSUM   +CA 1.000  S  1.000  RS 6.000  I322.000
H2 GAS    RS -2.000
H2S GAS   S  1.000  RS -2.000
HEMATITE  FE 2.000  RS 6.000
HYDROXAP +CA 5.000  P  4.000
ILLITE    K  0.600  MG 0.250  AL 2.300  SI 3.500
K-SPAR   +K  1.000  AL 1.000  SI 3.000
K-MICA   +K  1.000  AL 3.000  SI 3.000
K-MONT    K  0.330  AL 2.330  SI 3.670
KAOLINIT  AL 2.000  SI 2.000
LIGNITE  +C   1.0000RS -0.4000I1-25.0000I2  0.0000
MAGNESIT +MG 1.000  C  1.000  RS 4.000
MAGNETIT  FE 3.000  RS 8.000
MG.02CAL  C   1.0000CA  0.9800MG  0.0200
MIRABILI  NA  2.0000S   1.0000RS  6.0000
Mg-MONT   MG 0.167  AL 2.330  SI 3.670
Mg/Na EX  NA 2.000  MG -1.000
MgSiO3   +MG 1.000  SI 1.000
Mn(OH)3   MN 1.000  RS 3.000
MnO2      MN 1.000  RS 4.000
MnOOH     MN  1.0000RS  3.0000
N2 GAS    N  2.000  RS 0.000
NA-MONT   NA  0.3300AL  2.3300SI  3.6700
```

```
NACLINOP  NA 1.000  AL 1.000  SI 5.000
NH3 GAS   N  1.000  RS -3.000
NH4/CAEX  N  2.0000CA  -1.0000RS -6.0000
Na-MONT   NA 0.330  AL 2.330  SI 3.670
Na2SO4   +NA 2.000  S  1.000  RS 6.000
NaCl     +NA 1.000  CL 1.000
NaHCOL   +NA 1.000  C  1.000  RS 4.000
O2 GAS   +RS 4.000
PHLOGOPI  K  1.000  MG 3.000  AL 1.000  SI 3.000
PLAGAN38 +CA  0.3800NA  0.6200AL  1.3800SI  2.6200
PREHNITE +CA 2.000  AL 2.000  SI 3.000
PYRITE   -FE 1.000  S  2.000  RS 0.000  I3-30.00
RHODOCHR +MN 1.000  C  1.000  RS 6.000
SEPIOLIT  MG 2.000  SI 3.000
SIDERITE +FE 1.000  C  1.000  RS 6.000
STRENGIT  FE 1.000  P  1.000  RS 3.000
STRONITE  SR 1.000  C  1.000  RS 4.000  I8 0.000
SYLVITE   K   1.0000CL  1.0000
SiO2      SI 1.000
TALC      MG 3.000  SI 4.000
VIVIANIT  FE 3.000  P  4.000  RS 6.000
WITHERIT  BA 1.000  C  1.000  RS 4.000
*OTHER**
```

ATTACHMENT F:

Example of Data Report Printed by DB

```
Well name          : Richmond Hill

Owner              : City of Richmond Hill
                     Water Department
                     City Hall
                     Richmond Hill, GA  31324

Site ID            : 315620081190401
Latitude/longitude : 3156180811908
Date/time sampled  : 85/08/07 @ 0900

Temperature          23.8 degrees C   Fluoride              0.3 mg/l
Sp. Cond.            245.             Bromide              0.03 mg/l
pH                   8.00             Nitrite-Nitrate  <  0.010 mg/l as N
Eh                   N.D. volts       Total N (Kjd)         0.1 mg/l as N
Density               1.0 g/cm3       Phosphate           0.014 mg/l as P
Dissolved Oxygen      0.0 mg/l        Boron            <  0.020 mg/l
Alkalinity #        135.6 mg/l        Aluminum              0.0 mg/l
Calcium               32. mg/l        Silica               43.  mg/l as SiO2
Magnesium             5.0 mg/l        DOC                   3.4 mg/l
Sodium                12. mg/l        RS of DOC            N.D.
Potassium             1.6 mg/l        CH4 (aq)             N.D. mg/l
Barium              0.022 mg/l        Delta C-13          -7.05 o/oo
Strontium           0.310 mg/l        Carbon 14            1.79 % modern
Iron                0.130 mg/l        Delta S-34 (SO4)     N.D. o/oo
Manganese           0.006 mg/l        Delta S-34 (H2S)    -53.6 o/oo
Lithium             0.009 mg/l        Sr 87/86          0.70830
Chloride              4.8 mg/l        Delta Deuterium     -19.0 o/oo
Sulfate               5.7 mg/l        Delta O-18          -3.75 o/oo
Hydrogen Sulfide     0.09 mg/l        Tritium               0.6 TU

-----------------------------------------
N.D. = not determined
TDIC = Total Dissolved Inorganic Carbon
DOC = Dissolved Organic Carbon
Sp. Cond. = Specific Conductivity (uS/cm)
RS = Redox State
# = uncorrected, reported as HCO3-
```

☆ U.S. GOVERNMENT PRINTING OFFICE: 1991—673-739