UCRL-ID-130357

EQ3/6 Software Maintenance and Support Summary

Thomas J. Wolery

September 26, 1997

Lawrence Livermore National Laboratory

This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the Laboratory.
Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-ENG-48.

LLNL Yucca Mountain Project

Near-Field Environment Characterization Technical Area

EQ3/6 Software Maintenance and Support Summary

Deliverable No.: SP342FM4

Thomas J. Wolery

September 26, 1997

University of California

Lawrence Livermore
National Laboratory

EQ3/6 Software Maintenance and Support, Sept. 30, 1997

Summary

EQ3/6 is a software package for modeling chemical interactions in aqueous systems of geologic and engineering interest, such as water/rock, water/nuclear waste, and water/nuclear waste/rock. It is being used for a broad range of applications for the Yucca Mountain Site Characterization Project (YMSCP), including predictions of mineralogical changes in the altered zone, man-made materials investigations, and calculations of the long-term release of radionuclides from a variety of waste forms. Version 7.2a was the first qualified version of this software (certified on Aug. 17, 1994). Version 7.2b followed on Aug. 18, 1995 and is the most recent qualified version; it differs from version 7.2a only in that defects noted in the qualification report (Kishi, 7/12/94) were resolved. The present report describes the software maintenance and support activities that were carried out for the Version 7 line of the software in FY97. The most important of these activities is maintaining a system for the logging, documenting, and resolving software defects. This is required by the QARD (Supplement 1) in order for the software to remain certified. Other maintenance activities are necessary to retain functionality as computer hardware, operating systems, programming languages, and compilers change. In FY97, 12 software defects were logged and resolved, and two more were logged and awaiting resolution. These ranged in nature from the trivial to the serious. The corrected software will be released as version 7.2c in the first quarter of FY98. A version 8 line of totally rewritten code in modern Fortran, restructured to support new functionality, and with new capabilities for ion exchange modeling, pressure corrections, and redox disequilibrium, was completed to a baseline level in FY95. Subsequent activities required to complete qualification were not funded in FY96 and FY97. However, in FY97, each line of the software (version 7 and version 8) has been checked whenever a defect has been discovered in the other. A beta release of version 8.0 was made available to selected users in FY97. Qualification of version 8.0 may follow in FY98, depending on funding and direction. If this occurs, the certified software in the version 7 line will likely be retired and maintenance and support activities exclusively focused on the version 8 line.

1.0 Introduction

EQ3/6 and its usage have been extensively described in formal code documentation (Wolery, 1992ab; Daveler and Wolery, 1992; Wolery and Daveler, 1992). A more concise technical review describing typical applications is given by Wolery et al. (1990). Many other useful references may be found in these documents. The software is useful for both analyzing and predicting chemical interactions in systems containing water and other natural and man-made components such as rock, nuclear waste, and concretes and grouts. The applicability of the software is mainly subject to the availability of supporting thermodynamic and kinetic data. No geochemical modeling code or any similar chemical engineering code currently contains provision for all of the many possible discrete fundamental processes that may occur in a reacting aqueous system. However, EQ3/6 is one of the most advanced in this regard. Consequently, it is a fairly large and complex example of software. From a YMSCP perspective, the missing element of principal current interest is a set of capabilities to model the effects of boiling on geochemical processes, and vice versa. Some additional functionality in treating ion exchange modeling would also be highly desirable. Another missing element is the ability to include surface complexation modeling as part of the larger modeling of geochemical interactions.

The most recent versions of EQ3/6 (those produced since 1988) were developed exclusively to model geochemical processes of relevance to the YMSCP. Previously, the development of the software had also been supported by the Salt Repository Project; as a result of this, EQ3/6 retains capabilities for addressing the chemistry of strong brines. The range of applications of EQ3/6 within the YMSCP has been and remains broad, including studies pertaining to ambient groundwater geochemistry, waste form dissolution, interactions of released radionuclides with rock, rock/water interactions in the altered zone, degradation of concretes and grouts, and natural analogs of various sorts. In some activities, the software is used to design experiments or to analyze experimental results, in others, to predict processes that would occur in or about a repository over time scales that are too long to access experimentally. Various organizations participating in the YMSCP have used, are using, or will use this software, including LLNL, LANL, LBL, PNL, Intera, and IT Corp., as well as number of cooperating foreign organizations including NAGRA BRGM, and W.S. Atkins Environment. At the direction of the DOE, the software has also been provided to the NWTRB and the CNWRA. EQ3/6 has a large following outside the YMSCP, mostly in the academic community, where it is used in studies of such things as groundwater evolution, rock/water interactions, and the formation of ore bodies. This usage is encouraged, as it builds external confidence in the software.

EQ3/6 is a package which contains several code elements. EQ3NR is a speciation-solubility code. It is used for example to compute the speciation of ions and solutes in groundwaters, and the saturation indices which define the saturation states of such waters with respect to various minerals. EQ6 is a reaction-path code. It allows one to compute simulations of interactions as among groundwaters and minerals. EQPT is a data file preprocessor. XCON3 and XCON6 are input file reformatters for EQ3NR and EQ6, respectively. Although these are a convenience to code users, they also serve an important function in maintaining the test case library, which is itself another important element of the package. EQLIB is a supporting library. In Version 8, some parts of this have been spun off into two new libraries, EQLIBU and EQLIBG.

The first version of EQ3/6 to be certified for use in quality-affecting work was Version 7.2a. This was originally issued for use in non-quality-affecting work only on 12/28/93, and later certified on 8/17/94, upon the completion of an independent V&V (verification and validation) activity (Kishi, 7/12/94). The certification was restricted to use of the software on Sun SPARCstations, SGI workstations, and similar UNIX platforms (excluding Crays) and on 386/486/Pentium type PCs. Certification also excluded the use of the software for calculations involving solid solutions which are not ideal in either the molecular-mixing or the site-mixing sense (very little data exists for the excluded models).

The Software Verification and Test Report (Kishi, 7/12/94) described the independent V&V activity performed on Version 7.2a and the corresponding findings. Some minor errors were found, the most of them dealing with sign errors in equations in one part of the documentation, the EQ3NR user's manual (Wolery, 1992b). One potentially significant error was found in the coding for non-ideal solid solutions which was written up as an exclusion (for this and other reasons) from the certification. It was decided that Version 7.2a should be certified, and that the errors found should be resolved along with any others reported in time in a Version 7.2b, which would be "born certified." Until the Version 7 software is retired, the QARD (Supplement 1) mandates that a system be maintained for reporting and resolving reports of software defects. This activity was not carried out in FY96 because the YMSCP did not fund it. Funding for this activity was restored in FY97, narrowly averting the decertification of the software.

Prior to FY95, work had already been proceeding on the next development line, Version 8. Whereas Version 7 was written in a Fortran 77 (ANSI, 1978) incorporating many older features and programming styles, Version 8 had been rewritten in a much more modern Fortran, technically between Fortran 77 and Fortran 90 (ANSI, 1991). The Fortran 90 features used were restricted to those available as "extensions" in existing Fortran 77 compilers. This rewriting had also been accompanied by a significant restructuring of the software to allow the addition of new modeling capabilities. Three new major capabilities were added: a generic ion-exchange capability (with multiple exchangers and multiple-site extensions of the basic Gapon and Vanselow exchange models), thermodynamic pressure corrections, and the ability to model reaction-paths with redox disequilibrium. Several lesser new features were also added, including a user interface that is easier to work with, and a streamlined capability for making groundwater-mixing calculations. The Version 8 software is presently in beta testing by volunteer users. The software was completed to a baseline level in FY95, but the subsequent activities that were required to qualify this were not completed because of lack of funding of EQ3/6 in FY96, and of limited funding in F^97 to only maintain the certified software in the version 7 line. Version 8.0 will be qualified in FY98 if the appropriate funding and direction are provided. If this occurs, the version 7 line of code will probably be retired, and maintenance and support activities shifted exclusively to the version 8 line.

2.0 Code Maintenance

Code maintenance is taken here to refer to three general kinds of activities. One is the continued adaptation required to keep up with changes in hardware, operating system software, programming languages (including ANSI standards), and compilers. The second is the operation of a defect reporting and resolution system. The third is the construction of release packages of corrected code and their distribution to users. Other code activities (e.g., addition of new functional capabilities, or distribution of code with such new capabilities) are generally considered code development. The first and second of these activities may overlap; that is, if the software no longer works on a supported platform because of changes in the operating system, etc., this is logged as a defect.

The rise of Fortran 90 has had a significant effect on EQ3/6. As noted above, many Fortran 90 features were incorporated as "extensions" into Fortran 77 compilers even as the Fortran 90 standard was being developed. However, Fortran 90 compilers are now widely available. The Fortran 90 standard (ANSI, 1991) incorporates Fortran 77 in its entirety. Thus, a Fortran 90 compiler can be used to compile a Fortran 77 source code with no modification. On the other hand, the functionality of Fortran 90 is much greater. In particular, it establishes standard ways of carrying out certain functions, such as getting the time and date, that had to be obtained through non-standard means specific to given platform, operating systems, and compilers. Until FY97, we had resisted incorporation of "obligatory" Fortran 90 features (features which won't work with Fortran 77 compilers, due to the previous rarity of Fortran 90 compilers on Unix platforms (such compilers have been widely available for a couple of years now, but high cost deterred widespread installation). This year we added the first obligatory Fortran 90 feature to both the version 7 and 8 lines of code: the standard method of getting the date and time (this is discussed later in this report in connection with the correction of a potential year 2000 problem). If work with the version 8 software is supported in FY98 and beyond, we will add additional elements of "obligatory" Fortran 90. These will include the use of Fortran 90s run-time memory management, its greatly expanded set of intrinsic functions, and its new special capabilities for handling arithmetic operations involving floating-point vectors and matrices. This will allow simplifications and efficiencies in the software. The use of its run-time memory management capabilities, for example, would allow a way to avoid a repeat of V7 Defect 9702001 (discussed below).

The defect reporting and resolution system is based on the use of a Software Defect Log maintained as an electronic document using FrameMaker. This document is similar to other existing electronic documents, including a Software Backup Log, a Software Development Log, and a Software Distribution Log. The Development Log is at the heart of the configuration management system. It references the Defect Log when the software is modified to resolve defects, and also records all maintenance activities, as well as any development activities.

The defects described in the following section have been resolved in code that is being prepared for release as Version 7.2c. Actual distribution is expected in the first quarter of FY98.

3.0 Defect Reporting and Resolution

All defect reporting and resolution in the past year is associated with the Version 7 line. The Version 8 software has been checked for the possible existence of the same defects found for Version 7, and vice versa. Fourteen defect reports were logged in FY97, of which twelve were resolved. The other two were logged in late September and are still be resolved. Of the other twelve defects, two were discovered in FY96, and informally noted and fixed outside the QA system for use in non-YMSCP applications prior to the resumption of maintenance and support functions in FY97. All fourteen defects dealt with the software itself (as opposed to the documentation).

Each defect that is logged is assigned an identifying seven-digit number which contains the last two digits of the year, two digits identifying the month, with the last three digits being reserved for the defects in the corresponding time frame. Defects are also identified by version line. For example, V7 Defect 9706001 would be the first defect logged in June, 1997 for the version 7 line.

The typical EQ3/6 software defect causes a run to terminate prior to producing a full set of desired results (partial results may be obtained). EQ3/6 performs complex numerical operations requiring the use of iterative algorithms such as Newton-Raphson. A defect may lead to non-convergence, which may lead to an early termination of the run. In the course of trying to recover from such a situation, the EQ6 code will respond by trying to reduce the step size, and the run may terminate early because the step size has failed to exceed the minimum value for a predetermined number of steps. Another common cause of early termination in EQ3NR or EQ6 is the generation of an arithmetic exception such as a zero divide. Defects of this sort limit the usefulness of the software, but at least do not generate incorrect results which might be mistaken for correct results by the users. This means that users who encounter a defect usually know it. Defects which do produce incorrect results which could be taken for valid results, especially by experienced modelers, are the most dangerous. These are rare in the history of EQ3/6, but not unknown.

It is important to remember that EQ3/6 is a large and complex piece of software that has considerable breath of application. The logging of more than a dozen defects in one year does not mean that the software has a reliability problem. The instances in which users are adversely affected by these defects are not very common. This is in part because a single run typically only utilizes some of the many features of the software (but different runs typically utilize different combinations). In the writer's judgement, it is unlikely that any YMSCP calculations to date have been adversely affected by these defects, other than by the software exhibiting premature termination of runs (but see comments regarding V7 Defects 9703001 and 9703003)

V7 Defect 9610001: Unix shell scripts failed under Solaris 2.3. The Unix shell scripts RUNEQPT, RUNEQ36, and XCIF36, which are used as user interface codes to simplify running the EQ3/6 main codes on Unix platforms, contained constructs of the form, "@ fsize = ‘wc-l output‘[1]" to obtain the number of lines in a file (here "output"). These failed under Solaris 2.3 (discovered during an OS upgrade of Sun SPARCstations at LLNL). The scripts would stop due to an apparent syntax error. This defect was termed serious because it prevented the software from being used in the intended manner on Unix platforms. These constructs had to be broken up into ones of the form, "set wclout=‘wc -l output‘" ; @ fsize = $wclout[1] ; unset wclout" in order to get the scripts working again. This defect had been informally fixed on 3/18/96. There is no real standard for Unix scripting language analogous to the ANSI standards for Fortran and C. A possible solution to avoiding problems such as this would be to replace the Unix shell scripts with C or C++ programs. That is a potential future maintenance and support activity.

V7 Defect 9610002: The **nxridx** array in EQ6 was improperly dimensioned. This array was dimensioned using the wrong dimensioning parameter (**nxrtpa**, the maximum number of solid-solution reactants, instead of the correct **nrctpa**, the maximum number of reactants of any type). The incorrect dimension was too small, as **nxrtpa** can not be greater than **nrctpa**. This could have resulted in an array subscript out-of-bounds error. This defect was deemed moderately serious, although it is not known to have adversely affected any actual runs. In a production executable, integer data would have overwritten part of the character variable array **ureac**, in which the reactant names are stored. These are written on the output file in various tables as the calculation proceeds, so if this error actually ever manifested, it should have been obvious. Attempts to access the overflowed data via the **nxridx** array itself may have failed or succeeded, depending on the operating system and compiler. This defect was caught while inspecting source code for another problem. It had been informally fixed on 12/9/96.

V7 Defect 9612001: Occurrence of floating-point exceptions in solid solution calculations in subroutine hpsat.f. This defect was first reported Dr. Michael Siemann of the University of Clausthal. His group was attempting to develop solid solution models of evaporate minerals to account for substitutions at a fairly extreme trace level (example: Br^- for Cl^- in NaCl). Runs were terminating with a floating-point overflow. Initially, the problem was traced to an incorrect setup (by the Clausthal group) of the model parameters on the data file. However, when the data file problem was fixed, the software still exhibited problems. These manifested as floating point overflows or negative arguments to logarithms, depending on the specific problem being run (several examples were received from the Clausthal group). The Clausthal group was attempting to use a part of the solid solution modeling capability that had been specifically excluded from the software certification. However, the remaining problems lay within a more general part that was not excluded. This defect was therefore deemed serious. We note that trace level substitutions of radionclides in solid solutions are potentially of great significance to YMSCP performance assessment.

These remaining problems were traced to the hypothetical affinity calculations made in the EQLIB subroutine **hpsat.f**, and in its supporting function subroutine, **ssfunc.f**. The problems were threefold. First, the **hpsat.f** calculations involved defining one set of components as trace components (whose mole fractions are less than 1.0 x 10^{-8}) and non-trace components. Apparently the coding for the trace components had never been actually tested, as some necessary parts were actually absent from the subroutine. This allowed some variables to take on zero values by default, which then led to arithmetic exceptions. Second, the mole fraction values for neither the non-trace nor the trace components were checked against appropriate physical and computational limits. The iterative procedure in **hpsat.f** permitted mole fraction values slightly greater than one, or less than or equal to zero, to be returned by the algorithm. In testing, problems typically arose when a binary solid solution was dominated by one component. The iterated mole fraction value for this non-trace component was slightly greater than one. The calculated value for the other component (the trace component) was then calculated to be slightly less than zero. Third, the coding itself was less than optimal for avoiding floating-point problems. For example, the saturation index (SI) in **hpsat.f** and **ssfunc.f** was calculated by taking the logarithm of a complex product, rather than by taking a sum involving the logarithms of the corresponding quantities appear in that product.

These problems were fixed by appropriate rewriting of the **hpsat.f** and **ssfunc.f** subroutines, as discussed in the above analysis. The missing coding for trace components was added. Iterated mole fraction values were clamped between 1.0 and 1.0 x 10^{-14}.The potentially unsafe coding (which might have allowed floating point exceptions in intermediate stages of calculation) was replaced by safer coding.

On 12/19/96, a message was received from Paul Cloke (of the M&O, SAIC) via J. Wesley Davis concerning this same defect. Dr. Cloke was using the software for the Materials Disposition Project, analyzing the performance effects of disposing of excess fissile material in the potential Yucca Mountain repository.

V7 Defect 9702001: EQ3/6 was insufficiently dimensioned to handle the new (V8/R6) data files. The number of basis species and the number of various kinds of species (e.g., aqueous species, pure minerals, gas species) on the new V8/R6 data files now exceeds dimensioned limits in EQPT, EQ3NR, and EQ6. These codes stop with an error message on the first occurrence in which a dimensioned limit is exceeded. when the first such limit is exceeded. This defect was therefore deemed serious. The dimensioned limits associated with the problems were increased. In EQPT, the dimensioned limit for total species had been insufficient. In EQ3NR and EQ6, the limits for basis species, aqueous species, and pure minerals had all been insufficient. There was little safety margin in the case of other limits, such as for chemical elements and gas species (in all three codes). These limits were also increased.

Future growth in the size of the supporting data files is anticipated. A way to avoid this type of problem in the future would be to utilize the run-time memory management capabilities of Fortran 90. This may be a future maintenance and support activity.

V7 Defect 9703001: The use of the value "500." to mean "no data" is no longer safe. The value "500." was long used on the supporting data files and in the software to indicate a "no data" condition for log K parameters and associated quantities. This had been a safe practice because the log K values on the data files were always much less than this. However, some reactions on the newer (e.g., V8/R6) data files have log K values exceeding a value of 500. The reactions involved are all redox reactions. Most are associated with organic species. This was a serious problem. The code behavior (EQ3NR and EQ6) was not entirely consistent when a log K equaled or exceeded 500. In some instances, the species associated with the reaction was taken to be suppressed; in other instances, it is was not. Most YMSCP applications to date were probably not adversely affected, as the affected reactions are not important in most scenarios for which the software has thus far been used. This problem was fixed by replacing "500." as the "no data" value by "9999999." (seven nines).

Current data files still bear the "500." value to indicate "no data.". To accommodate this, the modified EQPT code presently replaces "500." values with "9999999." Further action is recommended to fix this remaining problem. In the meantime, a "real" value of 500 must be entered on a data file as 500.0001 or 499.9999 in order to be recognized as "real." There appear to be no examples of "real" values of 500 in the V8/R6 set of data files.

This defect is not thought likely to have adversely affected any YMSCP calculations made to date. Adverse effects are only likely if calculations have been made involving a strong role for organic species or certain odd species involving sulfur (S) or nitrogen (N). An example of concern would be EQ6 calculations involving interactions among radionuclides and organic ligands such as EDTA. The EDTA and its complexes might be inadvertently suppressed. However, we note that few if any of the affected organic species in the V8/R6 data file set would be considered strong complexing agents. Indeed EDTA is not represented. However, it remains possible that some users might have added it or something like to their own EQ3/6 data files.

V7 Defect 9703002: Reactant affinity crossover searches in EQ6 sometimes failed. There was a problem in EQ6 in that searches for reactant affinity crossovers (points at which these affinities change sign) sometimes failed and led to the step size being fixed at the minimum value, followed by premature termination of the run. This was therefore a serious problem. The problem was that this search was executed when the reactant status code jreac(nrc) was 0 (normal, reacting) or -1 (not reacting, at equilibrium). It should have been executed when jreac(nrc) was 0 (normal, reacting) or +1 (exhausted). When a reactant is at equilibrium, its affinity is technically zero. However, finite-difference representations may produce non-zero values. This defect led failed searches and thence to the premature termination. In the case of jreac(nrc) = +1, the search which should have been made was not made. This could have resulted in poor performance in locating the point at which an exhausted reactant should be reactivated, but would not have produced an erroneous calculation because EQ6 in general does not assume that the results of such finite-difference-based searches are sufficiently accurate. Subsequent checks are employed, which can send the code back to re-advance using a smaller step size. The appropriate changes were made in the **path.f** subroutine. Furthermore, checks were applied in subroutine **btaylr.f** to ensure that a finite-difference representation, if attempted for the jreac(nrc) = -1 case, would yield an exact zero value.

V7 Defect 9703003: Log K values were not updated after automatic basis switching in EQ6. There was a problem in EQ6 in that the log K values for the reactions were left unchanged when automatic basis switching was carried out. This was discovered in responding to a telephoned request for help from Frederic Gérard, a post-doc at LBNL. The switching mechanism properly changed the temperature function coefficients from which the log K values were to be calculated. However, the log K values themselves were left unchanged. This was a problem, since some of the reactions to which the log K values were supposed to correspond had changed. This was fixed by adding a call to the EQLIB subroutine **evdatr.f** in each of the two places that automatic basis switching can take place in EQ6, one in subroutine **eq6.f**, the other in subroutine **path.f**. These calls follow calls to the subroutine **pabssw.f**, which executes automatic basis switching.

In the writer's judgement, this defect is not likely to have adversely affected any YMSCP calculations made to date. The automatic basis switching feature is turned off by default, its use is not generally encouraged due to the higher computational overhead that would result in many if not most cases, and hence few users are likely to used this feature. If they did use it and the run did not crash, the adverse effect would be to produce highly bizarre results easily recognized by the average user. Anyone who might have used this feature is advised to re-run their problems, using either the uncorrected code but with automatic basis switching turned off, or with the corrected code. Note: there was no problem with the automatic basis switching in EQ3NR.

V7 Defect 9709001: The RUNEQPT script failed to detect EQPT run errors. If an EQPT error occurred (usually indicating a problem with the data file it was processing), the user was not immediately warned by RUNEQPT. Unless the user checked the EQPT output file for error messages, the first indication of a problem occurred when the user tried to run EQ3NR or EQ6 with the data1 file that was produced. This would be s incomplete, and the EQ3NR or EQ6 run would terminate prematurely with its own error message. This defect was inconvenient, but not serious. The necessary coding was simply missing from the script. It was present in the companion script, RUNEQ36. The missing coding was added to RUNEQPT.

V7 Defect 9709002: Year 2000 problem. The time and date stamping on EQ3/6 output files used only two digits to represent the year. Conceivably, output from a run made in 2095 could be confused with that from a run made in 1995. Also, it was noticed that in the present coding used to calculate the run time, the year 2000 would be misidentified as a leap year (years evenly divisible by 100 are not leap years). The run time would be incorrect for a run starting on or before Feb. 28, 2000 and ending on or after March 1, 2000. These problems were deemed not serious. The date stamping was modified to use four digits to represent the year. Thee leap year problem was also fixed. Fortran 90 coding to get the time and date was implemented, since this version of Fortran offers a standard way of doing this. The older methods of getting the time and date were retained as options for users who must still use Fortran 77.

V7 Defect 9709003: Wrong array (**uelemb**) initialized to blanks in XCON6. In the main program of XCON6 (**xcon6.f**), the **uelemb** array was initialized to blanks using the dimensions appropriate to the **uesrb** array. It is actually the latter array which should have been initialized. This problem was not serious. The number of elements in the **uesrb** array is generally greater than that in the **uelemb** array. Consequently, the blank initialization would proceed beyond the end of the array. However, no deleterious consequences have been noted (and if there were any, they would have been obvious). There is no harmful effect to the **uesrb** array not having been initialized to blanks. The initialization is basically just a matter of good practice. The correct array (**uesrb**) is now initialized to blanks

V7 Defect 9709004: Non-convergence in EQ6 due to a solid solution component with a very small mole fraction. Running the test case library input file **j13wtuff.6i** beyond the usual 200 step limit revealed a convergence problem when the code tried to precipitate the clinoptilolite solid solution. This led to the step size being cut to the minimum value, followed by early termination of the run. This defect is moderately serious. The problem is that on the V8/R6 com data file, the clinoptilolite solid solution now has an ammonium (NH4) end-member. Evidently this was not the case when Joan Delany did the original modeling on which this test case is based. The model system is maintained in an oxidizing condition by specification of a fixed atmospheric oxygen fugacity. Consequently, ammonia is pretty scarce in the system.

Under these conditions, the mole fraction of the ammonium end-member of the clinoptilolite would be far less than the machine epsilon (approximately 2.e-16 on a Sun SPARCstation). The number of moles would be extremely small ($< 1 \times 10^{-30}$ moles). Consequently, the number of moles of this end-member can not be used as a primary iteration variable (the end-member can't be added to the extended basis set) without producing a numerical singularity. Numerically, this is analogous to what happens when the number of moles of a pure mineral becomes very small. Then, however, the code simply drops the pure mineral from the phase assemblage and hence from the extended basis set. However, when a solid solution is in the phase assemblage, each of its unsuppressed end-members was put into the extended basis. There was no mechanism to have some end-members in the extended basis set, and others not. It wa apparent that such a mechanism was needed.

This problem could be overcome by the user by suppressing the offending end-member, or by deleting it from the data file. The problem with this solution was that conditions could change along the reaction path, such that the suppressed end-member should become significant. For example, in the J-13 water/tuff problem described above, if one removed the fixed fugacity constraint on the oxygen and added metallic iron as a new reactant (to simulate reaction of incoming modified groundwater with canister material), the nitrate in the groundwater would be chemically reduced, and the ammonium end-member of the clinoptilolite might become significant. Only a more knowledgeable code user would be likely to realize this and make the appropriate adjustments.

This problem was fixed as follows. A new **jkflag** status flag value of -11 was created for end-members which are not added to the extended basis set when the corresponding solid solution is precipitated. The corresponding numbers of moles and mole fractions are calculated from the corresponding mass action equations in an iterative expansion scheme in subroutine **ncmpz2.f**. Dynamic adjustments to the status of the end-members is made in subroutine **eqcalc.f**.

At the present time, an end-member is not added to the extended basis set only if its predicted mole fraction is less than 1 x 10^{-10}. Some experimentation was carried out using higher values. This work suggested that higher computational efficiency could be achieved by only having a small number of end-members (perhaps as few as one) in the extended basis set. This has the advantage that the size of the Jacobian matrix can be reduced, sometimes significantly. However, to achieve this, some changes will have to be made to the coding which writes the Jacobian matrix. In its present form, it does not correctly deal with cases in which end-members are not in the extended basis set but still have mole fractions greater than about 1.0 x 10^{-2}. This may form the basis for a future maintenance and support activity.

V7 Defect 9709005: EQPT failed to flag a species name mismatch when processing reactions read from a data file. If the name of a species in its associated reaction does not exactly match the name defined for the species at the top of its data block, EQPT should flag the condition, declaring an error. The code was not checking for this condition. The user would discover that there is a problem only later, when trying to run EQ3NR or EQ6. This is not serious. It is only an inconvenience in that the problem is apparent later. The requisite checks were added in subroutines **pcraq.f** (for aqueous species) and **pcrsg.f** (for mineral and gas species).

Two additional defects were logged as this report was being completed. V7 Defect 9709006 concerns the incorrect writing on the EQPT data1f output file of species triplets corresponding to certain Pitzer mu coefficients. This is trivial, as the correct triplet is written on the corresponding data1 file, which is what is actually used by EQ3NR and EQ6. The data1f file is only a tool for conveniently examining what is supposed to be on the corresponding data1 file (which is unformatted and not correctly viewable using a text editor). V7 Defect 9709007 concerns the writing of null characters on the EQ3NR and EQ6 output files when a table is written describing the coverage of the available Pitzer coefficients. This also is trivial. Neither of these defects has been completely resolved as of the writing of this report.

4.0 Release of Updates

No updates were released in FY97. Software corrected for defects discussed in this report is planned to be released as Version 7.2c in the first quarter of FY98. Normally we would expect to release one such whole version update per year. However, a number of defects appeared late in this fiscal year, so we decided to correct them before doing this release. Partial updates (patches) may be transmitted to selected YMSCP users between major updates (e.g., corrections to only one or more elements of the software package) if such users request them. This year, no such requests were received. Analyses of defects and suggestion of possible work-around measures have been provided in some instances and seem to have been sufficient until the Version 7.2c release is made.

5.0 Future Work

We will continue to maintain and support the version 7 line of code, at least until such time as a qualified version is produced in the version 8 line. Then we will probably retire the qualified versions in the version 7 line, and then focus all maintenance and support effort on the version 8 line (until such time as it, too, should be retired). If programmatic funding and direction oblige, we will add new functionality to the version 8 line of code. We will focus first on adding capabilities for modeling the effects of boiling on geochemical reactions, and vice versa. We will also continue to add some new functionality to the existing ion exchange modeling capability. After that, we may then provision for the treatment of surface complexation models. We plan to move the Version 8 line of code increasingly into Fortran 90.

References

ANSI, 1978,Progamming Language FORTRAN: ANSI X3.9-1978, ISO 1539-1980 (E): American National Standards Institute, New York.

ANSI, 1991, Fortran 90: ANSI X3.198-1991, ISO/IEC 1539-1991 (E): American National Standards Institute, New York.

Daveler, S. A., and Wolery, T. J., 1992, EQPT, A Data File Preprocessor for the EQ3/6 Software Package: User's Guide and Related Documentation (Version 7.0): UCRL-MA-110662-PT-II, Lawrence Livermore National Laboratory, Livermore, California.

Kishi, T., 1995, EQ3/6 Software Test and Verification Report 7/94: UCRL-ID-in press, Lawrence Livermore National Laboratory, Livermore, California.

Wolery, T. J., Jackson, K. J., Bourcier, W.L., Bruton, C.J., Viani, B.E., Knauss, K.G., and Delany, J.M., 1990, The EQ3/6 software package for geochemical modeling: Current Status: In Melchior, D.C., and Bassett, R.L., eds., *Chemical Modeling in Aqueous Systems II*, American Chemical Society Symposium Series, 416, American Chemical Society, Washington, DC, p. 104-116.

Wolery, T. J., 1992a, EQ3/6, A Software Package for Geochemical Modeling of Aqueous Systems: Package Overview and Installation Guide (Version 7.0): UCRL-MA-110662-PT-I, Lawrence Livermore National Laboratory, Livermore, California.

Wolery, T. J., 1992b, EQ3NR, A Computer Program for Geochemical Aqueous Speciation-Solubility Calculations: Theoretical Manual, User's Guide, and Related Documentation (Version 7.0): UCRL-MA-110662-PT-III, Lawrence Livermore National Laboratory, Livermore, California.

Wolery, T. J., and Daveler, S. A., 1992, EQ6, A Computer Program for Reaction-Path Modeling of Aqueous Geochemical Systems: Theoretical Manual, User's Guide, and Related Documentation (Version 7.0): UCRL-MA-110662-PT-IV, Lawrence Livermore National Laboratory, Livermore, California.