DI: UCRL-MA-110662 PT IV
SCR: LSCR198
MI: 30084-M04-001 (Addendum)
December 11, 1998
QA: L

MOL.19990920.0169

SOFTWARE QUALIFICATION REPORT (SQR)

ADDENDUM TO EXISTING LLNL DOCUMENT UCRL-MA-110662 PT IV:

IMPLEMENTATION OF A SOLID-CENTERED FLOW-THROUGH MODE FOR EQ6 VERSION 7.2B

CSCI: UCRL-MA-110662 V 7.2b
SCR: LSCR198
MI: 30084-M04-001 (Addendum only)

Prepared by:

Harlan Stockman — Date: 12-14-98

Reviewed by:

Kaveh Zarrabi — Date: 12-15-98

Concurred by:

Robert A. Morgan, Engineering Assurance — Date: 12/23/98

Approved by:

Peter Gottlieb, Supervisor
Degraded & External Criticality Department — Date: 12/23/98

Table of Contents

Description — Page

1.0 SUMMARY 1

1.1 DESCRIPTION OF EQ3/6 7.2B BASIC PACKAGE 1
1.2 SOFTWARE NEEDS NOT MET BY ORIGINAL EQ6 PACKAGE 2
1.2.1 Need for Solid-Centered Flow-Through Mode 2
1.2.2 Need for Compact Output Files 2
1.2.3 Need for Informative Screen Output 3
1.3 SUMMARY OF FUNCTIONALITY SUPPLIED BY THE ADDENDUM TO EQ6 3
1.3.1 Solid-Centered Flow-Through Mode 3
1.3.2 Compact Output Files 3
1.3.3 Informative Screen Output 4
1.3.4 Array Size Changes 4
1.3.5 Identification as Altered Version of EQ6 4
1.4 SOFTWARE VALIDATION PROCESS 4
1.4.1 Validation Methods 4
1.4.2 Implementing Documents 5
1.4.3 Acceptance Criteria 5
1.4.4 Additional Software Documentation 5

2.0 INSTALLATION 6

2.1 HARDWARE AND OPERATING SYSTEM REQUIREMENTS 6
2.2 SUPPLEMENTAL MEDIA 6
2.3 INSTALLATION PROCEDURE 6
2.3.1 Install over Previous EQ6.EXE 7
2.3.2 Keep old EQ6.EXE, and Install `Eq6new` in a Separate Directory 7
2.4 INSTALLATION TESTS AND RESULTS 7

3.0 SOFTWARE VALIDATION 8

3.1 COMPARISON WITH ANALYTICAL SOLUTIONS (INDEPENDENT TEST CASE) 8
3.2 COMPARISON WITH `ALLPOST`/`NXTINPUT` CALCULATIONS (INDEPENDENT TEST CASE) 10
3.3 COMPARISON AGAINST STANDARD EQ6 BENCHMARK SUITE 14
3.4 TEST OF OUTPUT FROM `HWS410` AND `HWSTXT` SUBROUTINES 15

4.0 INSTRUCTIONS FOR USING NEW SOLID-CENTERED FLOW-THROUGH MODE 22

5.0 RECOMMENDATIONS 25

5.1 SOFTWARE ELEMENTS VALIDATED 25
5.2 SOFTWARE ELEMENTS NOT VALIDATED 25
5.3 RECOMMENDATIONS FOR SOFTWARE QUALIFICATION 25

6.0 REFERENCES 26

List of Attachments

Attachment I. Source Code Changes (15 pages)

Attachment II. Test Case Checklist (3 pages)

Attachment III. Concatenated `fc` (File Compare) Outputs for Section 3.3 Tests (26 pages)

1.0 SUMMARY

1.1 Description of EQ3/6 7.2b Basic Package

The EQ3/6 software package originated in the mid-1970s at Northwestern University [1]. Since 1978, Lawrence Livermore National Laboratory (LLNL) has been responsible for maintenance of the package. Most recently, EQ3/6 been maintained under the sponsorship of the Civilian Radioactive Waste Management Program of the U.S. Department of Energy. The major components of the EQ3/6 package include: EQ3NR, a speciation-solubility code; EQ6, a reaction path code which models water/rock interaction or fluid mixing in either a pure reaction progress mode or a time mode; EQPT, a data file preprocessor; EQLIB, a supporting software library; and several (>5) supporting thermodynamic data files. The supporting data files contain both thermodynamic standard state and activity coefficient-related data. Most of the data files support the use of the Davies or B-dot equations for the activity coefficients; two others support the use of Pitzer's equations. The temperature range of the thermodynamic data varies from 25 °C only for some species, to a full range of 0-300 °C for others. EQPT takes a formatted data file (a `data0` file) and writes an unformatted `data1` file, which is read by EQ3NR and EQ6. EQ3NR is useful for analyzing groundwater chemistry data, calculating solubility limits and determining whether certain reactions are in states of partial equilibrium or disequilibrium. EQ3NR is also required to initialize some EQ6 calculations.

Only the EQ6 code was modified for this report (minor changes were made to the `eqlib.for` source, but per QAP-SI-0, Computer Software Qualification, revision 4, section 3.34, those changes do not constitute "modifications"). EQ6 calculates the interaction of an aqueous solution and a set of "reactants", and the code can model fluid mixing and the consequences of changes in temperature. The code operates both in a pure reaction progress frame and in a time frame. In a time frame calculation, the user specifies rate laws for the progress of the irreversible reactions. Otherwise, only relative rates are specified. EQ6 uses a hybrid Newton-Raphson technique to make thermodynamic calculations. This method is supported by a set of algorithms that create and optimize starting values. EQ6 uses an ordinary differential equation (ODE) integration algorithm to solve rate equations in time mode. The codes in the EQ3/6 package are written in FORTRAN 77 and have been developed to run under the MS-DOS/Windows and UNIX operating system on platforms ranging from workstations to supercomputers. Further information on the codes of the EQ3/6 package is provided in Refs. 2, 3, and 4.

EQ6 is used by the Yucca Mountain Project Management and Operating Contractor (hereafter called "M&O") to calculate the degradation of waste packages (WPs) in water. These calculations are performed in support of internal and external criticality assessments, and for estimation of radionuclide transport through drifts and into rock fractures. The aqueous fluid in these calculations typically begins with a composition similar to J-13 well water[5]. The "reactants" are the components of the WPs (steels, fuel, criticality control materials such as gadolinium phosphate, high level waste (HLW) glass), or the engineered and natural materials external to the WPs (volcanic rocks, fracture fillings, backfill, and concrete). In addition, J-13 water may itself become a reactant, added continuously throughout the calculation.

As provided by LLNL, EQ6 principally models batch processes, operating in a closed system. The code has a mode for a "fluid-centered flow-through system", but that mode has no simple analogy with the processes involved in the degradation of WPs.

Throughout this report, a `bold courier` font is used to denote computer file names, parameters in programs or input files, or operating system (OS) commands. OS commands are indented by at least one space, and the "Enter" key implicitly follows each OS command.

1.2 Software Needs Not Met by Original EQ6 Package

1.2.1 Need for Solid-Centered Flow-Through Mode

In the proposed repository at Yucca Mountain, there is a small probability that waste packages (WPs) may be disrupted by rock falls, such that breaches, or holes, develop in the top surface of the WP [6]. Water may then drip into the packages, eventually filling the available void volume. In packages containing significant amounts of fissile materials, the water may act as a moderator, enhancing the likelihood of criticality. If the dripping is continuous, there is a possibility that water will overflow from the package, and the continuous dilution will remove neutron-absorbing components (such as gadolinium) that were originally added for criticality control. Some fissile materials may also be flushed from the system, possibly to reconcentrate outside the WP. The simulation of such processes, with a chemical reaction-path code, requires a "solid-centered flow-through" mode (p 44, Ref. 4). In this mode, an incremental amount of "fresh" water enters the WP system in each time step, displacing an equivalent volume of water out of the system.

EQ6 version 7.2b, as distributed by LLNL, does not contain a solid-centered flow-through mode. To add this mode, it is necessary to change the `eq6.for` source code, and recompile the source. However, by using a variant of the "special reactant" type built into EQ6, it is possible to add the functionality of a solid-centered flow-through mode in a very simple and straightforward manner.

1.2.2 Need for Compact Output Files

The amount of output generated by EQ6 can be enormous. EQ6 users at the M&O currently simulate a solid-centered flow-through system with the "`allpost/nxtinput`" method [7]. This method chains many EQ6 runs together, concatenates the output files, then post-processes and condenses the intermediate output to save space. The `allpost/nxtinput` method was developed to avoid changing or amending the EQ6 source code; thus the results are qualified according to LLNL's existing procedures. However, even with the space-saving measures, a single `allpost/nxtinput` run typically generates 300 MB of output, and a series of runs for a particular WP may generate well over 2 GB. The constant archival of results, and the clearing of outputs from the system, offer many chances for error and data loss. The resultant output files are somewhat tedious to process; blocks of output are pasted into Excel files, with many more hand operations and chances for error. The generation of Excel plots from a single run can take days. The binary output and tab-delimited files proposed in this Software Qualification Report (SQR) rarely exceed 40 MB, contain approximately ten times as much information as the files generated by the older method, and are easily accessed by graphical post-processors.

1.2.3 Need for Informative Screen Output

As distributed by LLNL, EQ6 version 7.2b gives no indication of the status of a calculation (apart from reports of errors or convergence failures) until the calculation is completed. Since the complex calculations performed by the M&O may take hours to complete, yet may also "hang" and become caught in infinite loops, there is a tendency for users to become impatient and terminate a calculation prematurely, unless there are signs of progress.

1.3 Summary of Functionality Supplied by the Addendum to EQ6

Of the additions made to the `eq6.for` source, only the first (Solid-Centered Flow-Through Mode, section 1.3.1) will "affect the calculational structure or flow ... and mathematical algorithms", per QAP-SI-0 rev 4, section 3.34. The other code additions allow production of compact output files, enhance the information printed to the screen during long runs, enhance the code capabilities for WP modeling by increasing array sizes, or simply print an additional notice identifying the addendum.

For simplicity, the modified source files are referred to hereafter as `eq6new.for` (the replacement for `eq6.for`) and `eqlibnew.for` (the replacement for `eqlib.for`). It must be noted that the changes made to `eqlib.for` comprise minor fixes to output or date formats, and array sizes, and do not constitute true modifications, per section 3.34 of QAP-SI-0 revision 4. All the modifications are documented in the `eq6new.for` and `eqlibnew.for` source files. The headers for the source files, and the modified sections of code, are reproduced in Attachment I.

1.3.1 Solid-Centered Flow-Through Mode

The new mode is induced with a "special-special" reactant (`jcode=5`, Daveler format term `DISPLACER`). The `jcode` number `5` is immediately trapped and converted to `jcode=2`, with a `jhws` flag set to `1` (all other jcodes have `jhws=0`). Apart from the input trapping, the distinction between the `DISPLACER` and `SPECIAL` reactants is seen only in one 9-line block of the `reacts` subroutine, when the total moles of elements in the rock+water system (`mte` array) is adjusted by adding in the `DISPLACER` reactant, and subtracting out a commensurate amount of the total aqueous elements (`mteaq` array). This adjustment is made in change 12 of Attachment I.

1.3.2 Compact Output Files

The subroutine `hws4io` produces a binary file called `hwsdata`. That file contains the aqueous solution moles and molality of dissolved elements, the number of moles of each solid (including solid solutions), saturation states, gas fugacities, reactant moles, pH, ionic strength, reaction progress and physical time (if run in kinetic mode) for every time step. The `hwsdata` file can be read by the `PP` program described in Ref. 8, or by short C programs that run in stand-alone mode. The companion subroutine `hwstxt` produces tab-delimited files that are easily pasted into spreadsheets (the files `elem_aqu.txt`, `elem_min.txt`, `elem_m_a.txt` and `elem_tot.txt`). These files are designed to provide the input necessary for neutronics calculations, and require very little additional manipulation, reducing the chance of user error. While the files produced by `hwstxt` are text, the routine uses binary writes to overcome the limits of FORTRAN 77 formatted I/O.

1.3.3 Informative Screen Output

At each calculation step, `eq6new.for` writes the step number (and log time in seconds, if in kinetic mode) to the screen. Whenever a change in the solid phase assemblage is encountered, the solids in the new assemblage are printed as well. These additions to screen output are found in routines `hws4io` and `modexz`, in changes 6 and 8 of Attachment I.

1.3.4 Array Size Changes

As described in the EQ3/6 V&V report[9], it is necessary to increase the dimensions for several of the static arrays in EQ3/6. The changes are made through parameter statements, and do not constitute "modifications" per QAP-SI-0 revision 4 section 3.34. The most important of these changes in the number of "special reactants" allowed for the code. The WPs modeled by the M&O are complex, and may involve many "reactants" in the form of alloys and fuel components; up to 12 special reactants have been used in runs to date. All the array size changes are indicated in the header of the `eq6new.for` file, and in Attachment I.

1.3.5 Identification as Altered Version of EQ6

After printing the normal LLNL disclaimers, `eq6new.for` prints the following message to the screen and to the `output` file:

```
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-M04-001 for this Addendum only.
 CRWMS M&O; original document # UCRL-MA-110662.
 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

This notice is the only change `eq6new.for` makes to the standard `eq6.for output` file. The message is printed by the subroutine `hwsdis`.

1.4 Software Validation Process

Two parts of the validation process are distinguished here. The first is the ***initial validation*** presented in section 3.0, which involves a detailed analysis to establish the accuracy of the flow-through mode implemented in `eq6new.for`. The second is a ***test procedure for installation*** of the addendum on individual CPUs (the checklist for the latter process is provided in Attachment II).

1.4.1 Validation Methods

The ***initial validation*** process, described in section 3.0, involved four parts:
(1) Comparison of `eq6new.for` results against analytical solutions for a passive tracer in a solid-centered flow-through system, to validate and measure the accuracy of the new functionality.

(2) Comparison of `eq6new.for` results against a run produced by the `allpost/nxtinput` method [6,7], to demonstrate the accuracy of the new functionality, for a complex problem typical of those encountered in WP calculations at the M&O.

(3) Comparison of `eq6new.for` results with those achieved by the LLNL distribution of `eq6.for` (version 7.2b), for the standard EQ3/6 test suite, to verify that the new functionality in `eq6new.for` does not interfere with the original capabilities of the `eq6.for` program.

(4) Proof that the supplemental I/O functions of the addendum (specifically, files produced by routines `hws4io` and `hwstxt`) produce outputs that agree with the `tab` (or `*.6t`) and `output` (or `*.6o`) files of the normal EQ3/6 7.2b distribution.

The ***test procedure for installation,*** described in Attachment II, is used to verify that the software addendum has been installed correctly on individual PCs.

1.4.2 Implementing Documents

Implementing documents for the addendum include the EQ6 manual [4], and the current document. Section 4.0 of the current document describes how to use the new solid-centered flow-through mode.

1.4.3 Acceptance Criteria

For numerical results, an acceptance criterion of ±10% is used for the two independent test cases (sections 3.1 and 3.2), and a criterion of ±1% is used for the comparison against the standard EQ6 test cases (sections 3.3 and 3.4). For test cases with graphical output, a visual comparison between actual and expected results will be used as acceptance criteria. Visual comparisons that do not match the expected results will require documented justification of acceptance for software validation.

1.4.4 Additional Software Documentation

No additional software documentation is required. Section 2.0 of this report documents installation of the addendum, and section 4.0 gives brief instructions for the use of the new capabilities.

2.0 INSTALLATION

2.1 Hardware and Operating System Requirements

The requirements for `eq6new` are essentially the same as those for `eq6`, as described in Ref. 9. The program described herein runs in a DOS window under Windows 95 and Windows 98, on processors compatible with the Intel 486, Pentium, Pentium Pro, and Pentium II families. However, because flow-through calculations typically involve many time steps, and because the calculations performed by the M&O tend to be complex and to involve many components, performance may be unacceptable with the minimal hardware described in Ref. 9. For typical WP calculations, a minimum system includes a 166 MHz Pentium or Pentium/MMX processor, and 32 MB RAM. The output files can be quite large in flow-through calculations, with binary output files (`hwsdata` files) ranging from 10 to 50 MB, so the user should have at least 1 GB free hard drive space, and a method to archive and remove unused files from the system. However, because the `eq6new.for` solid-centered flow-through mode tends to run fast, the binaries can be recreated quickly if only the input files (which are rarely greater than 40 KB) and the corresponding thermodynamic databases are saved.

2.2 Supplemental Media

The installer must obtain a copy of the supplemental media from Software Configuration Management. The supplemental media for `eq6new` contain two self-extracting executable files, `eq6new1A.exe` and `eq6new1B.exe`. The first file (`eq6new1A.exe`) contains the executable program `eq6new.exe` (a replacement for `eq6.exe`), the five test cases (`.6i` files), and the comparison output files (`.6nw` files) required for the Test Case Checklist (Attachment II). The second file (`eq6new1B.exe`) contains the source code for `eq6new.for` and `eqlibnew.for`, the additional test cases `new_1.6i` and `new_2.6i` described in section 3.1, and the `f01_2022.6i` and `f01_2022_alln.6i` test cases described in section 3.2.

The `eq6new.exe` file was compiled with the Lahey LF90 compiler, version 3.5, with the command line:

```
LF90 eq6new.for eqlibnew.for -pack -bind -tp.
```

2.3 Installation Procedure

This installation procedure requires that EQ3/6 version 7.2b has already been installed on the PC, per Ref. 9. It is assumed that the installer is familiar with the opening and operation of a DOS console.

Begin the installation of the supplemental media by opening an MS-DOS console on the target CPU. A directory called `eq6new` is then created on the target CPU, and the supplemental media files `eq6new1A.exe` and `eq6new1B.exe` are copied into the that directory. The Supplemental Media files are then executed, which produces the uncompressed files in the directory `eq6new`. Assuming the Supplemental Media disk is in the `A:` drive, the process is accomplished with the following DOS commands:

```
cd \
md eq6new
```

```
cd \eq6new
copy a:eq6new1A.exe
copy a:eq6new1B.exe
eq6new1A
eq6new1B
```

at the DOS prompt. The "Enter" key must follow each of the DOS commands above.

From this point, two installation methods are possible, *viz*:

2.3.1 Install over Previous EQ6.EXE

Locate the directory containing the `eq6.exe` binary; this is the directory indicated by the `SET EQ36CO` line in the `autoexec.bat` file. Enter this directory, and rename `eq6.exe` to `eq6old.exe`. Copy the `eq6new.exe` binary into the same directory, and rename it `eq6.exe`. The `runeq6` program will now launch the new executable, and EQ6 can be used as before, and the program will recognize the EQ3/6 environment variables, allowing one to launch the program from any directory. This procedure MUST be used to test the installation per Attachment II. After the installation test, `eq6old.exe` can be copied back into `eq6.exe`, and `eq6new` can be run from a separate directory, as described in section 2.3.2.

2.3.2 Keep old EQ6.EXE, and Install `Eq6new` in a Separate Directory

Copy the appropriate `data1.*` file (*e.g.* `data1.com` or `data1.hmw`)from the EQ3/6 database directory (as indicated by the `SET EQ36DA` line in the autoexec.bat file), into the `eq6new` directory. Rename the `data1.*` file to `data1`. Each time a `*.6i` file is to be run, it must first be copied to a file named `input`, and `input` must reside in the `eq6new` directory. The program `eq6new` is then invoked in this directory (without arguments), and the standard EQ6 output is directed to a file named `output`, and the compact binary file is named `hwsdata`.

2.4 Installation Tests and Results

The test procedure described here is modified after Attachment II of Ref. 9. It is presumed here that there is a prior installation of EQ3/6 on the system, per Ref. 9. After `eq6new.exe` is copied over the prior version of `eq6.exe`, the additional test cases described in Attachment II (Test Case Checklist) must be run for each installation. After completion of the test cases, a signed copy of Attachment II must be returned (for each target CPU) to the Software Configuration Manager, accompanied by a memo signed by the installer (or installers') responsible manager.

3.0 SOFTWARE VALIDATION

Sections 3.1 and 3.2 exercise the new solid-centered flow-through mode, added to EQ6 to create `eq6new.for`. Section 3.3 shows that the addition of the new mode does not affect `eq6new`'s accuracy in running the 17 test cases in LLNL's distribution of EQ3/6. Section 3.4 shows that the new output modes of the addendum produce results that agree with the default EQ3/6 output modes.

3.1 Comparison with Analytical Solutions (Independent Test Case)

This test case compares `eq6new.for` calculations against the analytical solution for flushing of an inert tracer (that is, a chemical component that does not enter into a discrete solid phase). The files `new_1.6`, `new_2.6i` and `new_01.6i` perform the test with the EQ6 parameter `dlzmx2` (maximum step size, in units of reaction progress `zi`) set to 0.1, 0.2 and 0.01, respectively. Each of these `6i` input files uses the pickup file generated by the standard EQ6 test case `micro.6i` (with 1 Liter solution, muscovite mica and K-feldspar present), and titrates in up to 100 "moles" of water, with a "mole" of water defined to equal 1 Liter (111.0173631 moles hydrogen, and 55.50868156 moles oxygen). The tracer is chlorine, initially present at $1.00 \cdot 10^{-4}$ Molal. It is simple to show that if chlorine does not enter into the minerals, its concentration C will follow:

$$dC = C \cdot (-dV_{added}/V_{total}) \rightarrow C/C_0 = exp(-\Delta V_{added}/V_{total}) \rightarrow \qquad (1)$$

$$ln(C/C_0) = -\Delta V_{added}/V_{total},$$

where C_0 is the initial concentration, ΔV_{added} is the added volume of the `DISPLACER` reactant, and V_{total} is the fixed volume of the aqueous system.

Figure 1 shows the Molality of chlorine (log scale), plotted against $\Delta V_{added}/V_{total}$, for the `eq6new` calculation with `dlzmx2` = 0.01 (`new_01.6i`). The plot shows the linear behavior expected from equation 1; after the system has been flushed ~100 times, the chlorine concentration has reached extremely low levels, $< 10^{-47}$ Molal. Figure 2 shows the difference between the analytical solution in equation 1, and the results of the `eq6new` runs, expressed as % error. For `dlzmx2` = 0.01, the error is well under 1%, even after the Cl concentration has been reduced to $\sim 10^{-47}$ Molal. This is an extremely harsh test of the flush-through algorithm, for two reasons: (1) in most cases, `eq6new` will be used for calculations in which solute concentrations are buffered by solids in the system, which tends to lower error; and (2) the EQ6 `dlzmx2` parameter is based on the sum of reaction progress for all reactants, not just the displacer reactant. In the `new*.6i` test cases, the displacer reactant dominates reaction progress, so a `dlzmx2` of 0.2 implies up to 20% of the system volume can be added (as displacer reactant) at any time step. When other reactants are present, the maximum amount of a displacer (added per time step) will typically be substantially less than `dlzmx2`. Since the errors range from <1%

for the case with `dlzmx2=0.01`, to < 9% `for dlzmx2=0.2`, the results are within the ±10% criteria established in section 1.4.3 for the independent test cases.

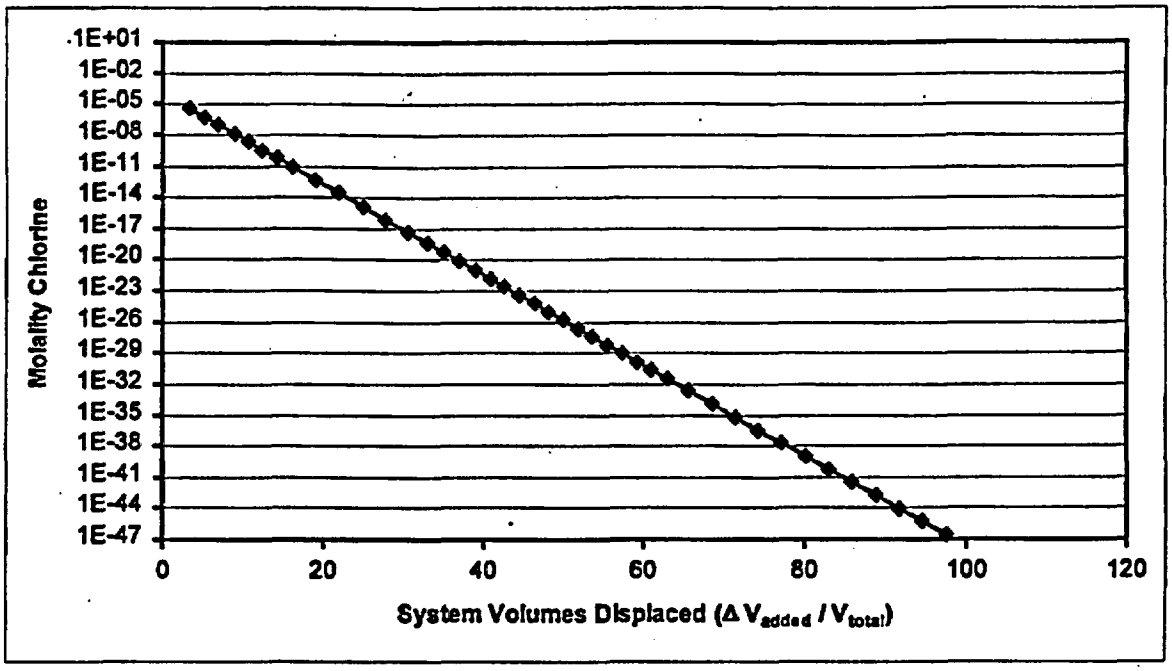

Figure 1. Molality of Chlorine Calculated for Test Case `new_01.6i`, Showing Exponential Behavior Expected for Conservative Tracer (Note Log Scale on Ordinate).

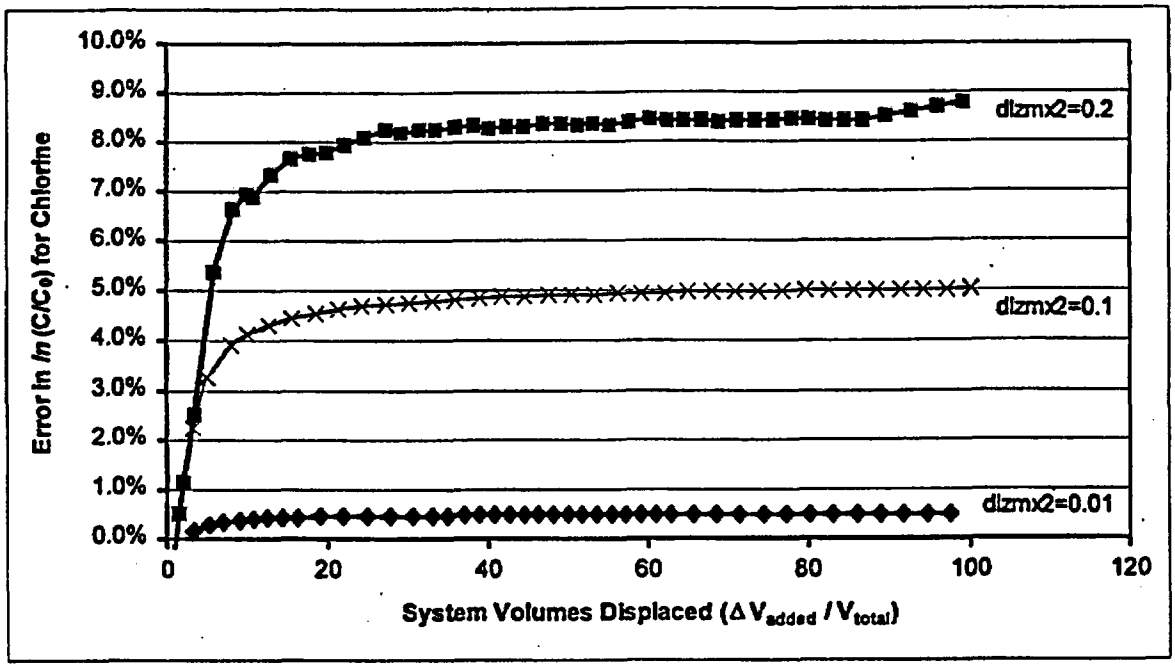

Figure 2. Error in Predicted Molality of Chlorine, Relative to Equation 1, for Three Different Values of the Parameter `dlzmx2`.

3.2 Comparison with `allpost/nxtinput` Calculations (Independent Test Case)

The `eq6new` results may also be compared against calculations made with the `allpost` and `nxtinput` programs [7]. The `allpost/nxtinput` programs consist of a series of C and/or batch-language drivers that chain together many separate EQ6 runs, to simulate the flush through system. Within each EQ6 run, J-13 water and other reactants are added to the system; between successive EQ6 run, the amount of water and solutes in the system is adjusted to maintain an approximately constant volume of one Liter. The `allpost/nxtinput` method has been verified by independent hand calculations [7], and the codes have been ported to the PC[10].

A calculation for degradation of WPs, containing Fast Flux Test Facility (FFTF) fuel, provides a comparison between `eq6new` and the PC version of `allpost/nxtinput`. The calculation begins with first-stage runs `F01_2204.6i` and `F01_2204_alln.6i` for `eq6new` and `allpost/nxtinput`, respectively (these files are in the supplemental electronic media). In the first stage, J-13 water is reacted with the HLW glass and steels outside the DOE Spent Nuclear Fuel (SNF) canisters, at a high drip rate (0.5 m^3/year into a package with ~6 m^3 of void space), for a period of ~3770 years. The package void volume is flushed ~324 times in this process, removing much of the alkalinity in the glass. In the second stage (file `F01_2202.6i` for `eq6new`, `F01_2022_alln.6i` for `allpost/nxtinput`) the J-13 water drip rate into the package is reduced to 0.015 m^3/year, and the SNF canister is "breached", allowing the fuel and remaining components to interact. The package is flushed through with J-13 water corresponding to a minimum of 400 times the package volume. This is a harsh comparison, since there is plenty of opportunity for the two methods to lose synchronicity between the two

stages, as the two methods choose steps sizes differently, and the calculation involves tens of thousands of `eq6` time steps. (`eq6new` tends to pack more time steps into regions where concentrations change rapidly, to minimize errors and provide greater coverage of "interesting" areas of the calculation, where solid minerals are forming or disappearing.)

Figures 3 through 6 compare the Gd, pH, U and Pu aqueous concentrations calculated by the two methods, for the second stage. The continuous line indicates the `eq6new` results, and the black x's denote the results obtained with `allpost/nxtinput`. The agreement between the two methods is excellent, despite the choice of very different time steps. For the plots with a linear concentration scale, the disagreements between the `eq6new` and `allpost/nxtinputpeak` concentrations are less than 5%, again within the ±10% criterion established in section 1.4.3. The pH range in figure 4 (5.4 ≤ pH ≤ 7.65) is typical of expected repository conditions.

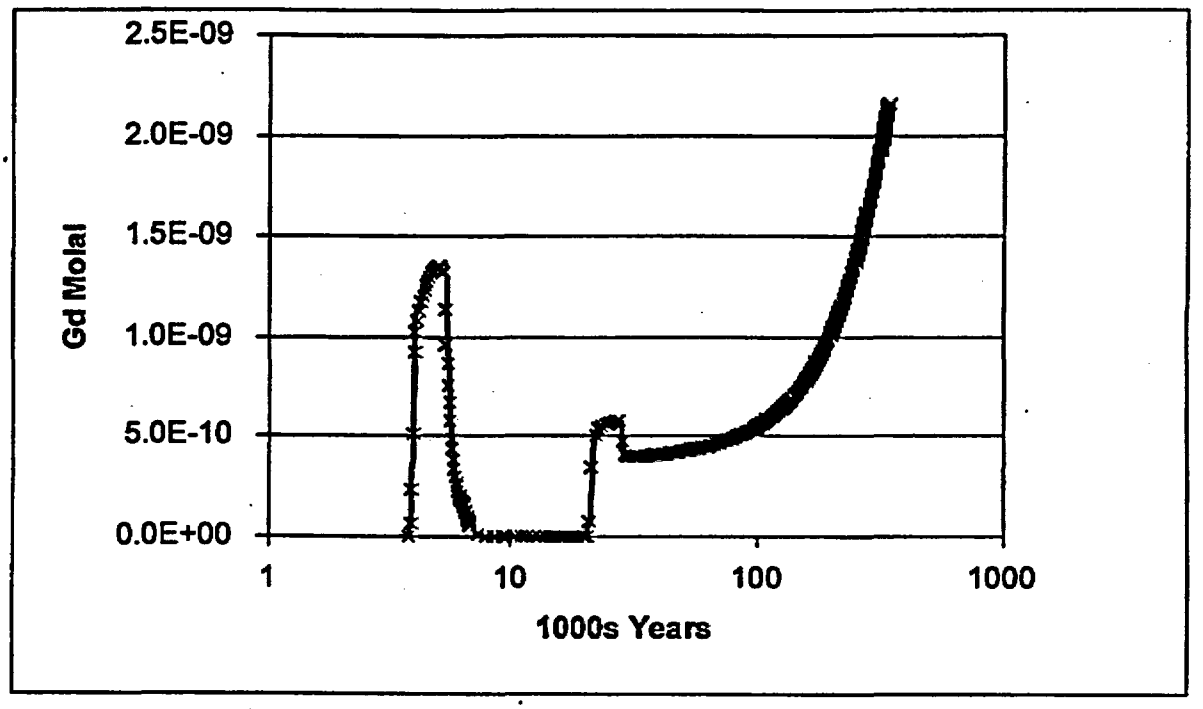

Figure 3. Comparison of `eq6new.for` Calculations for Gd Molality (Continuous Line) and `allpost/nxtinput` Results (Crosses).

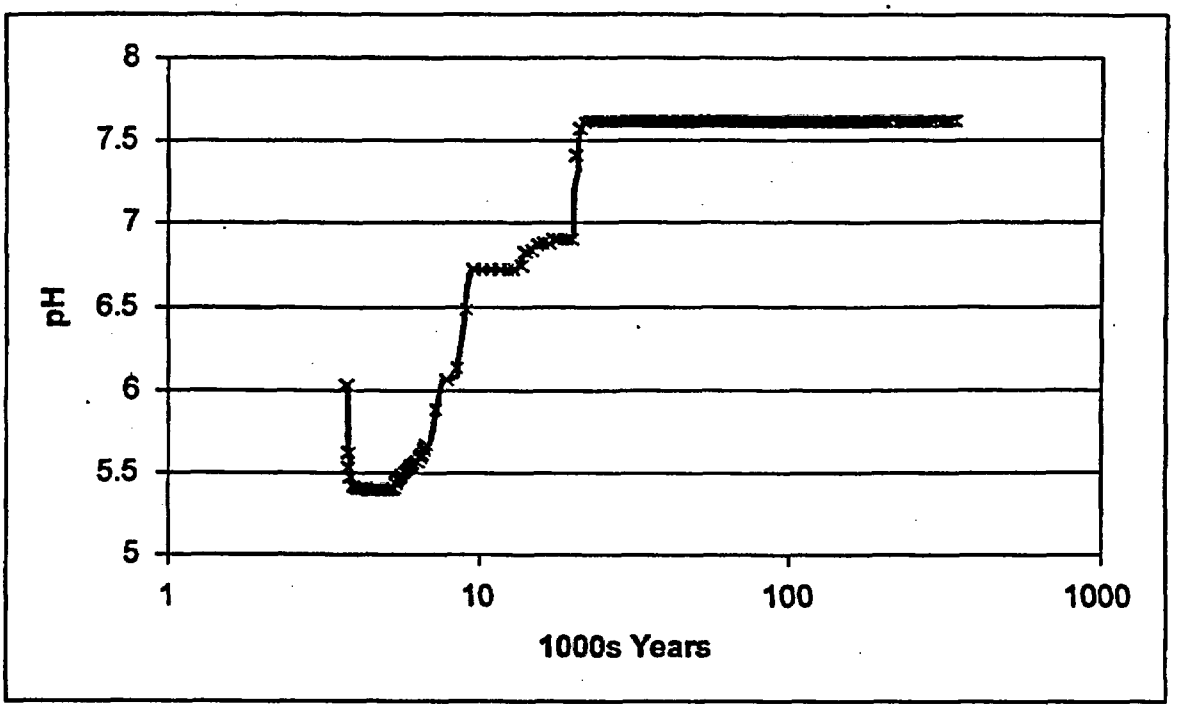

Figure 4. Comparison of `eq6new.for` Calculations for pH (Continuous Line) and `allpost/nxtinput` Results (Crosses).

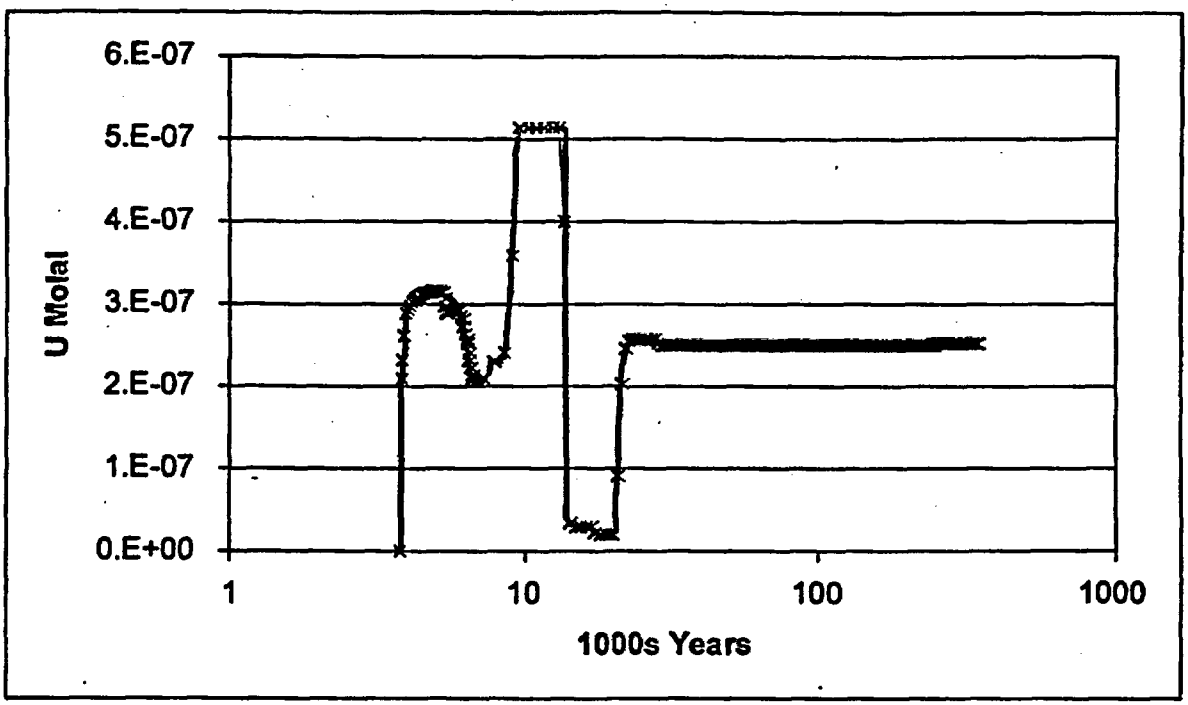

Figure 5. Comparison of `eq6new.for` Calculations for U Molality (Continuous Line) and `allpost/nxtinput` Results (Crosses).

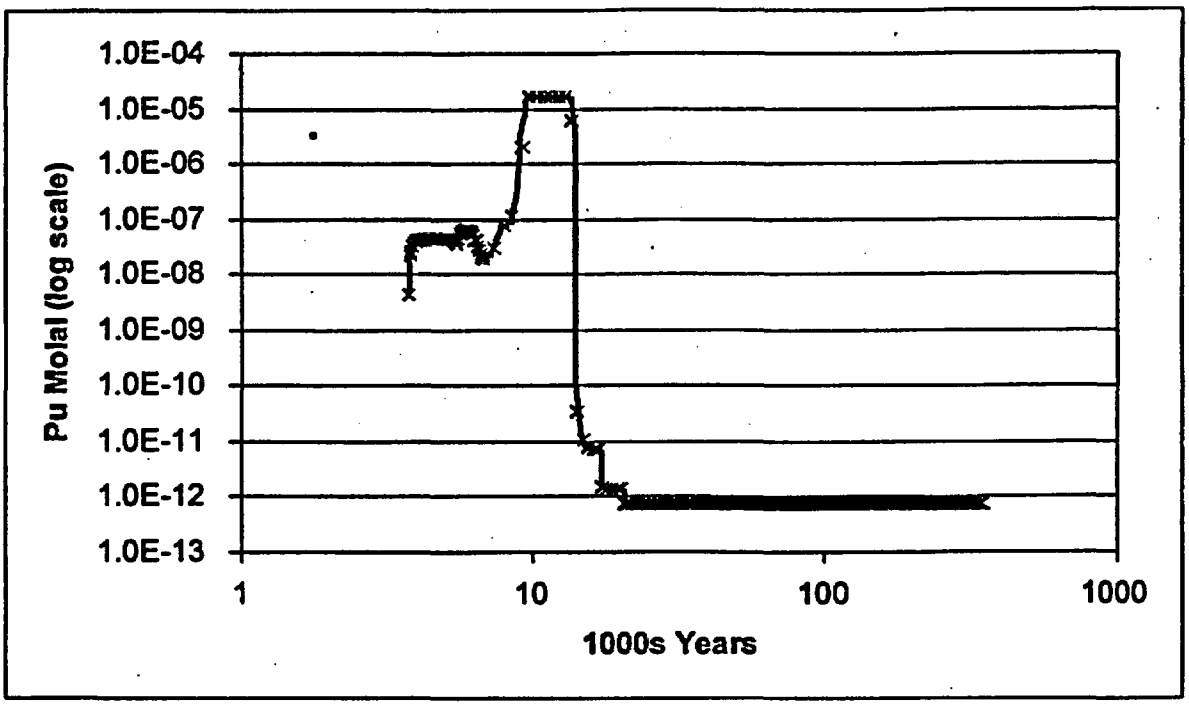

Figure 6. Comparison of `eq6new.for` Calculations for Pu Molality (Continuous Line) and `allpost/nxtinput` Results (Crosses).

3.3 Comparison against Standard EQ6 Benchmark Suite

In sections 3.1 and 3.2, it was shown that the added code correctly produces a solid-centered flow-through calculation. However, it is important to show that the added code does not interfere with any other functionality of EQ6. To show that there is no interference, the standard EQ6 test suite can be run for both the modified code (`eq6new.for`) and the standard version of `eq6.for` from the EQ3/6 version 7.2b distribution. The output files from the two codes are then compared and checked for differences. To minimize differences that might result from changes in compiler libraries and optimization strategies, both `eq6new.for` and `eq6.for` (and the corresponding libraries `eqlibnew.for` and `eqlib.for`) were recompiled with the exact same options and same version of the Lahey compiler (Lahey LF90 Version 3.5). The compiler command lines were:

```
LF90  eq6new.for eqlibnew.for -pack -bind -tp
LF90  eq6.for eqlib.for -pack -bind -tp
```

where the `-tp` switch optimizes for the Pentium, `-pack` compresses arrays to minimize the executable size, and `-bind` attaches the 32-bit DOS extender to the executable. Seventeen standard EQ6 test cases were run: `DEDOLO.6i`, `HEATQF.6i`, `J13WSF.6i`, `J13WTUFF.6i`, `METHANE.6i`, `MICRO.6i`, `MICROFT.6i`, `PPTCALA.6i`,

`PPTQTZ.6i`, `PPTQTZA.6i`, `RWTITR.6i`, `EVSWGYHA.6i`, `CALHAL.6i`, `FWBRMIX.6i`, `GYPNACL.6i`, `GYPSUMB.6i`, and `CRISQTZ.6i`. The comparison uses the MS-DOS `fc` (file compare) command (the concatenated difference files from `fc` are given in in Attachment III). In brief, the `fc` comparison found no differences between each set of files, other than the expected differences in date, dimensioned array sizes, version name ("Pentium" *vs.* P5), and the header added by the `hwsdis` routine. The numerical difference of 0% is within the 1% acceptance criterion established in section 1.4.3. Thus the added flow-through mode appears to have no effect on the normal functionality of EQ6.

3.4 Test of Output from `hws4io` and `hwstxt` Subroutines

This test uses the tab file `micro.6t` (created by EQ6 test case `micro.6i`), and compares the results against the files created by the added subroutines `hws4io` and `hwstxt`: files `hwsdata`, `elem_aqu.txt`, `elem_min.txt`, `elem_m_a.txt` and `elem_tot.txt`. The elemental concentrations (Molality) of the aqueous phase were extracted from `hwsdata` with the `PP` program[8], version 1.10, and saved by `PP` as tab-delimited text. Table 1 shows the comparison of the `hws4io`/`PP` results with those from `micro.6t`, for the same set of time steps; the agreement is excellent (`PP` saves one more significant digit). The slight differences arise from the lower precision saved by EQ3/6 in creating tab files (this difference is most obvious for zi and pH values). In addition, the EQ3/6 tab file (`micro.6t`) saves $\log_{10}$(Molality) to 4 places only, and additional error is created on performing the $10^{\log10(\text{Molality})}$ operation to obtain true Molality. Figures 7 and 8 show the `PP` output as imported into Excel, with the results from `micro.6t` superposed; figures 9 and 10 are the corresponding plots produced by `PP`, after importing the `micro.6t` results into `PP`. The correspondence between the two figures is obvious. In effect, this comparison also validates the use of `PP` version 1.10 to extract data from the `hwsdata` file.

Table 1. Comparison of zi, pH, Concentrations and (Molality) from `micro.6t` File vs. `hws4io` Output Imported into Excel 97.

zi (6t)	zi (PP)	pH (6t)	pH (PP)	Al (6t)	Al (PP)
0.00000E+00	0.00000E+00	4.00000	4.00000	1.00000E-12	1.00000E-12
1.00000E-08	1.00000E-08	4.00020	4.00017	1.00000E-08	1.00010E-08
1.00000E-07	1.00000E-07	4.00170	4.00168	1.00000E-07	1.00001E-07
1.00000E-06	1.00000E-06	4.01710	4.01713	1.00000E-06	1.00000E-06
7.13400E-06	7.13403E-06	4.13870	4.13867	7.12853E-06	7.12853E-06
1.00000E-05	1.00000E-05	4.14820	4.14817	6.69268E-06	6.69203E-06
5.86000E-05	5.86016E-05	4.41140	4.41144	1.20365E-06	1.20357E-06
8.78100E-05	8.78119E-05	4.88490	4.88494	6.93426E-08	6.93458E-08
1.00000E-04	1.00000E-04	5.92940	5.92945	1.22321E-09	1.22309E-09
1.01900E-04	1.01897E-04	7.36760	7.36762	4.33311E-09	4.33357E-09
1.49600E-04	1.49576E-04	7.36660	7.36657	2.94781E-09	2.94772E-09
Cl (6t)	Cl (PP)	K (6t)	K (PP)	Si (6t)	Si (PP)
1.01158E-04	1.01163E-04	1.00000E-12	1.00000E-12	1.00000E-12	1.00000E-12
1.01158E-04	1.01163E-04	1.00000E-08	1.00010E-08	2.99985E-08	3.00010E-08
1.01158E-04	1.01163E-04	1.00000E-07	1.00001E-07	2.99985E-07	3.00001E-07
1.01158E-04	1.01163E-04	1.00000E-06	1.00000E-06	2.99985E-06	3.00000E-06
1.01158E-04	1.01163E-04	7.13346E-06	7.13402E-06	2.14042E-05	2.14021E-05
1.01158E-04	1.01163E-04	1.00000E-05	9.99999E-06	2.99985E-05	3.00000E-05
1.01158E-04	1.01162E-04	5.86003E-05	5.86013E-05	1.75711E-04	1.75691E-04
1.01158E-04	1.01162E-04	8.78213E-05	8.78115E-05	1.75711E-04	1.75692E-04
1.01158E-04	1.01162E-04	1.00000E-04	9.99995E-05	1.99986E-04	2.00000E-04
1.01158E-04	1.01162E-04	1.01906E-04	1.01896E-04	2.03798E-04	2.03798E-04
1.01158E-04	1.01162E-04	1.02141E-04	1.02142E-04	2.98951E-04	2.98976E-04

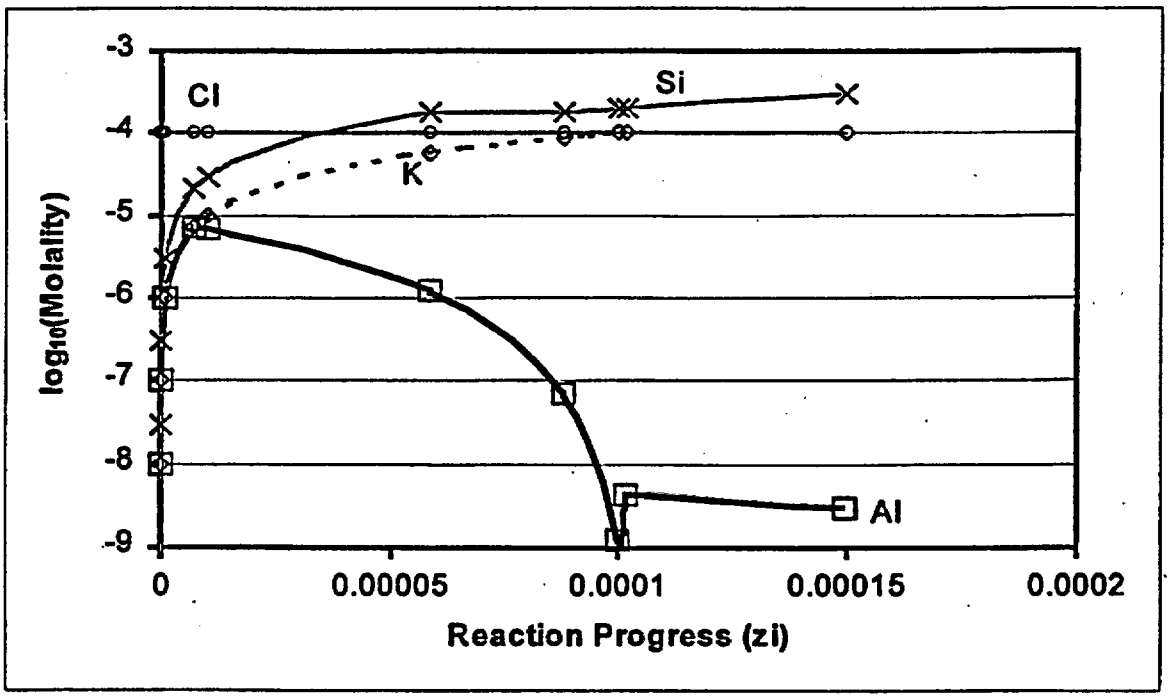

Figure 7. Excel 97 Comparison Plot of Elemental Concentrations. Solid Lines: `hwsdata` File Imported into `PP v1.10`, Exported as Tab-Delimited Text, Then Imported into Excel 97. Discrete Markers: EQ3/6 Tab File `micro.6t`, Imported into Excel 97.

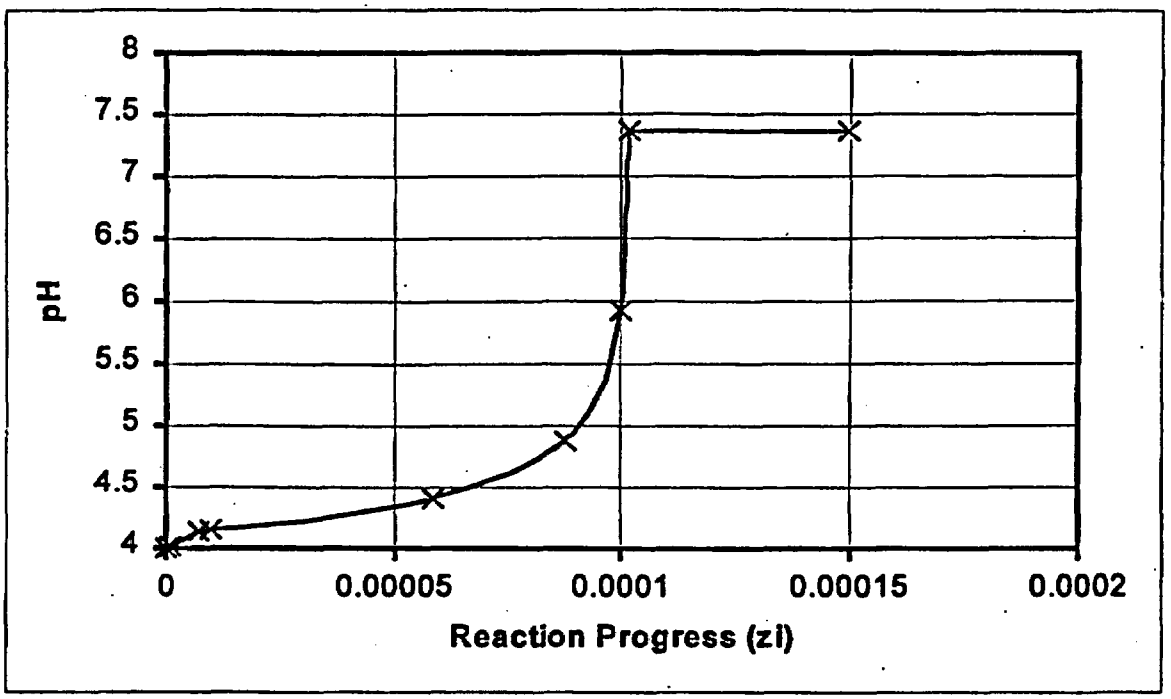

Figure 8. Excel 97 Comparison Plot of pH. Solid Lines: `hwsdata` File Imported into `PP v 1.10`, Exported as Tab-Delimited Text, Then Imported into Excel 97. Discrete Markers: EQ3/6 Tab File `micro.6t`, Imported into Excel 97.

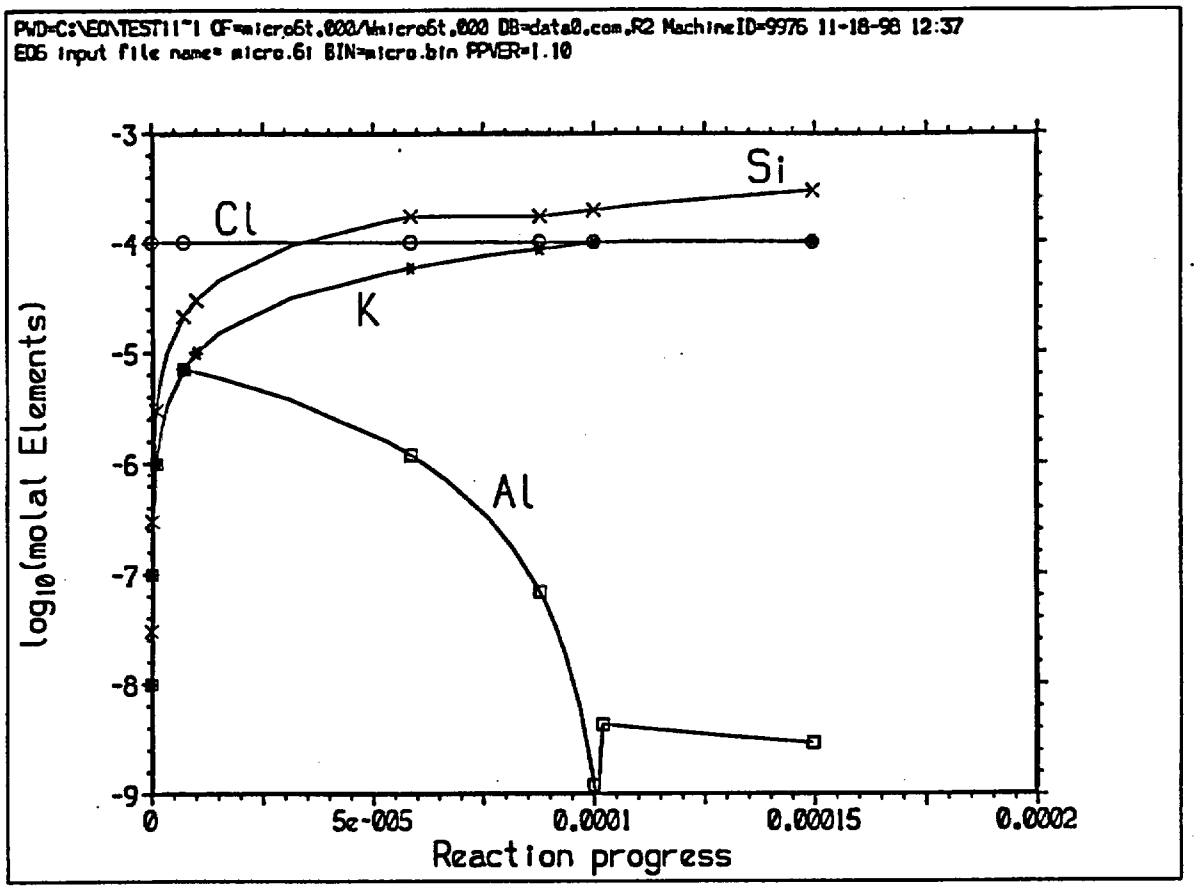

Figure 9. PP Comparison Plot, Elemental Concentrations. Solid Lines: hwsdata File Imported and Plotted in PP v 1.10. Discrete Markers: EQ3/6 Tab File micro.6t, Imported into Excel 97, Then Exported as Tab-Delimited Text, Then Imported and Plotted in PP. Compare to Figure 7.

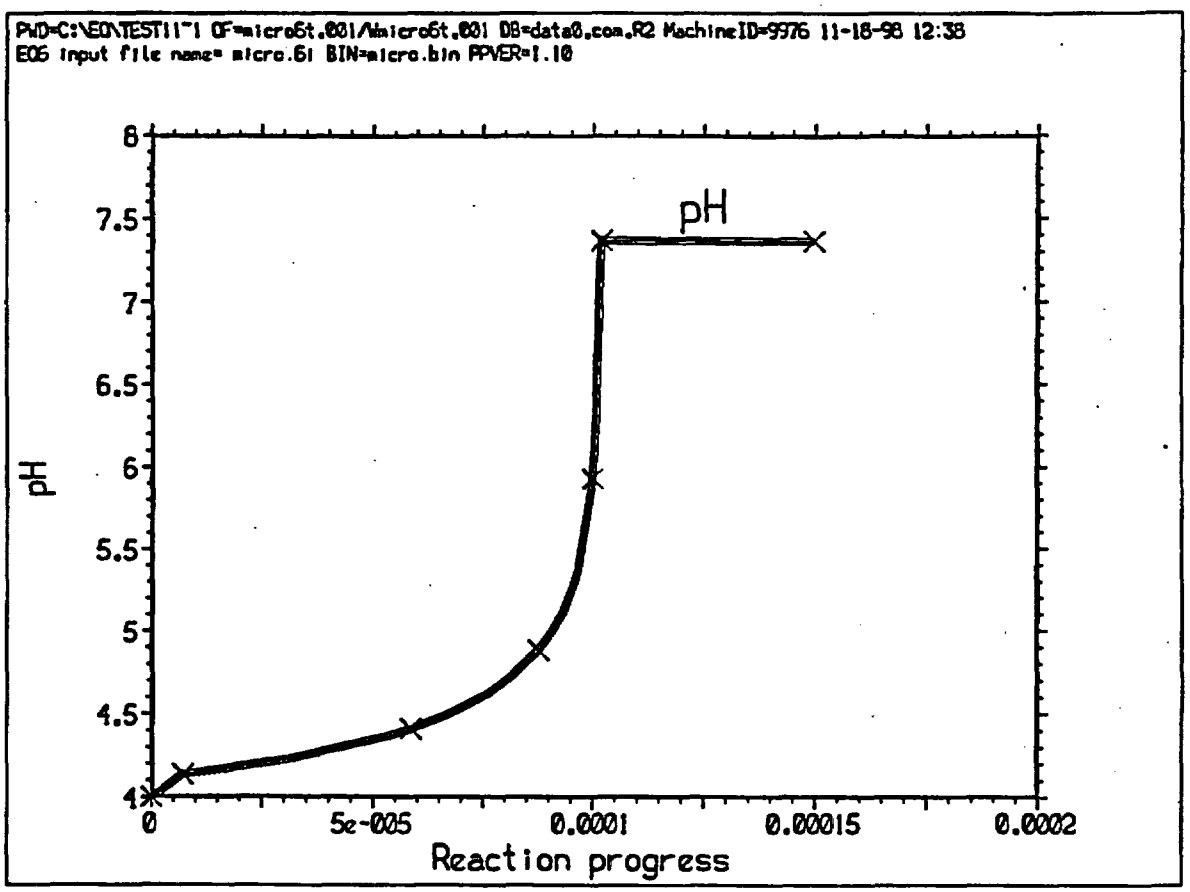

Figure 10. PP Comparison Plot, pH. Solid Lines: hwsdata File Imported and Plotted in PP v1.10. Discrete Markers: EQ3/6 Tab File micro.6t, Imported into Excel 97, Then Exported as Tab-Delimited Text, Then Imported and Plotted in PP. Compare to Figure 8.

Table 2 gives the `elem_aqu.txt` file created by the `hwstxt` subroutine; comparison against the results in Table 1 shows good agreement. Table 3 compares the calculation of mineral volumes in `elem_min.txt` against the `micro.6t` tab file results (called "`cc cre`" in the `micro.6t` file). Again, the agreement is good.

Table 2. Portion of `elem_aqu.txt` File (Compare Against Table 1).

```
This xls file gives MOLES in aqueous phase
data0.com.R2
|EQ6 input file name= micro.6i
|Description= "Microcline dissolution in pH 4 HCl"
```

Years	zi	pH	Eh	fO2	IonStr
0.0000E+00	0.0000E+00	4.0000E+00	-9.9900E+02	0.0000E+00	1.0116E-04
0.0000E+00	1.0000E-08	4.0002E+00	-9.9900E+02	0.0000E+00	1.0119E-04
0.0000E+00	1.0000E-07	4.0017E+00	-9.9900E+02	0.0000E+00	1.0144E-04
0.0000E+00	1.0000E-06	4.0171E+00	-9.9900E+02	0.0000E+00	1.0396E-04
0.0000E+00	7.1340E-06	4.1387E+00	-9.9900E+02	0.0000E+00	1.2069E-04
0.0000E+00	1.0000E-05	4.1482E+00	-9.9900E+02	0.0000E+00	1.1946E-04
0.0000E+00	5.8602E-05	4.4114E+00	-9.9900E+02	0.0000E+00	1.0423E-04
0.0000E+00	8.7812E-05	4.8849E+00	-9.9900E+02	0.0000E+00	1.0130E-04
0.0000E+00	1.0000E-04	5.9294E+00	-9.9900E+02	0.0000E+00	1.0119E-04
0.0000E+00	1.0190E-04	7.3676E+00	-9.9900E+02	0.0000E+00	1.0194E-04
0.0000E+00	1.4958E-04	7.3666E+00	-9.9900E+02	0.0000E+00	1.0219E-04

O	Al	Cl	H	K	Si
5.5508E+01	1.0000E-12	1.0116E-04	1.1102E+02	1.0000E-12	1.0000E-12
5.5508E+01	1.0001E-08	1.0116E-04	1.1102E+02	1.0001E-08	3.0001E-08
5.5508E+01	1.0000E-07	1.0116E-04	1.1102E+02	1.0000E-07	3.0000E-07
5.5508E+01	1.0000E-06	1.0116E-04	1.1102E+02	1.0000E-06	3.0000E-06
5.5508E+01	7.1285E-06	1.0116E-04	1.1102E+02	7.1340E-06	2.1402E-05
5.5509E+01	6.6920E-06	1.0116E-04	1.1102E+02	1.0000E-05	3.0000E-05
5.5509E+01	1.2036E-06	1.0116E-04	1.1102E+02	5.8602E-05	1.7569E-04
5.5509E+01	6.9346E-08	1.0116E-04	1.1102E+02	8.7812E-05	1.7569E-04
5.5509E+01	1.2231E-09	1.0116E-04	1.1102E+02	1.0000E-04	2.0000E-04
5.5509E+01	4.3336E-09	1.0116E-04	1.1102E+02	1.0190E-04	2.0380E-04
5.5509E+01	2.9477E-09	1.0116E-04	1.1102E+02	1.0214E-04	2.9898E-04

Table 3. Comparison of Mineral Volumes from `elem_min.txt` and `micro.6t` (Tab File).

zi (Reaction Progress)	Minerals Volume (cm^3) from `elem_min.txt`	Minerals Volume (cm^3) from `micro.6t`
0.0000E+00	0.0000E+00	0.0000E+00
1.0000E-08	0.0000E+00	0.0000E+00
1.0000E-07	0.0000E+00	0.0000E+00
1.0000E-06	0.0000E+00	0.0000E+00
7.1340E-06	9.7645E-08	9.7640E-08
1.0000E-05	5.8749E-05	5.8750E-05
5.8602E-05	1.0230E-03	1.0230E-03
8.7812E-05	4.3661E-03	4.3660E-03
1.0000E-04	4.9759E-03	4.9760E-03
1.0190E-04	5.0702E-03	5.0700E-03
1.4958E-04	1.0873E+02	1.0870E+02

4.0 INSTRUCTIONS FOR USING NEW SOLID-CENTERED FLOW-THROUGH MODE

The simplest way to use the new functionality is to modify an existing EQ6 input file. The existing EQ6 input file should represent a system with 1 Liter of aqueous solvent. The displacing solvent (*e.g.*, J-13 water) is added as a reactant, described by either `jcode= 5` (Wolery or W format files), or "`| reactant type | DISPLACER`" (Davelar or D format files). The composition of the displacer reactant is defined so that 1 "mole" of the reactant corresponds to the initial volume of aqueous solvent in the system (1 Liter, in this case). The `dlzmx2` parameter should be set to 0.2 or lower (see section 3.1). When the input file is run, the system aqueous volume will stay approximately constant. The rate at which the system is flushed will be determined by the `rk1` (in the EQ6 input file) for the `DISPLACER` (or `jcode= 5`) reactant. The files `new_01.6i`, `new_1.6i`, `new_2.6i` and `F01_2022.6i` can be used as examples to set up a calculation. (Note the file `F01_2022_alln.6i` is for the `allpost/nxtinput` formalism, and is NOT to be used with the displacer mode.)

If the results of the calculation are to be "picked up" to begin a new run, the `jcode= 5` notation (W format) or "`| reactant type | DISPLACER`" (D format) must be reset in the pickup file.

An `eq6new` run produces five files, in addition to the files produced by the LLNL distribution of EQ3/6. These files are called: `elem_aqu.txt`, `elem_min.txt`, `elem_m_a.txt`, `elem_tot.txt`, and `hwsdata`. The four "elem" files are easily pulled into Excel spreadsheets (Excel versions 5.0 through Excel 97) by two methods: (1) use the Excel File "open" dialog, specify text file, then select "delimited", "tab-delimited", then select "finish". This first process creates a new Excel workbook. (2) Alternatively, each "elem" file can be read into a separate sheet of an existing Excel file with the following procedure: (a) open an existing Excel file with 4 blank sheets; (b) open each "elem" file in an ASCII text editor, such as WordPad; (c) in the text editor, use the "Edit – Select All" dialog to mark the entire file; (d) type Ctrl-C (or use the editor "copy" dialog; either process copies the entire file to the windows buffer); (e) place the cursor in the upper left corner of a blank Excel sheet; (f) type Ctrl-V (or use the Paste dialog); (g) repeat for each elem file, using a blank sheet in Excel. The `hwsdata` file can be read with a program such as `PP`[8]; the format of the `hwsdata` binary is given in the `PP` manual.

The `elem_*.txt` files all have similar format. For example, the `elem_aqu.txt` file contains a descriptive header, then a block of (6+*N*) columns by *nstep* rows, where *N* is the number of elements in the system, and *nstep* is the number of time steps selected for output (the `hwstxt` routine prints a step at each EQ6 print point, thus it typically prints about every tenth step, with higher step densities in time intervals that have rapid chemical changes). A small portion of a typical `elem_aqu.txt` file is below; the first line is a reminder of the file content;

the second records the thermodynamic database used for the calculation; and the 3rd and 4th give the first two lines of the input file, to identify the source of the calculation.

```
This xls file gives MOLES in aqueous phase
data0.nuc.R8
|EQ6 input file name= F01_2022.6i                                    |
|Description= React J-13 water with metals in an FFTF waste package.   |
Years        zi           pH           Eh           fO2          IonStr       O            Ag           Al
 3.7637E+03  3.7611E+02  6.0328E+00  8.6215E-01  2.0999E-01  1.2770E-02  5.5524E+01  1.4914E-06  1.3484E-06
 3.7637E+03  3.7611E+02  6.0328E+00  8.6215E-01  2.0999E-01  1.2770E-02  5.5524E+01  1.4914E-06  1.3490E-06
 3.7679E+03  3.7616E+02  5.8859E+00  8.7084E-01  2.0999E-01  2.4875E-02  5.5536E+01  1.5863E-06  3.4303E-06
 3.7883E+03  3.7642E+02  5.6844E+00  8.8277E-01  2.0999E-01  8.1151E-02  5.5592E+01  1.8509E-06  1.6833E-05
 3.8124E+03  3.7673E+02  5.5955E+00  8.8803E-01  2.0999E-01  1.4433E-01  5.5654E+01  2.0385E-06  1.9482E-05
```

The first six columns give the Years, reaction progress variable (zi), pH, Eh, fO_2 and ionic strength; the remaining columns give the moles of each element in the aqueous phase.

A small portion of an `elem_min.txt` file is shown below. The format is similar to the format for the aqueous phase, but this table contains (4+*N*) columns; note the 4th column is an estimate of the total volume of minerals. The volume estimate excludes any fictive "fixed fugacity" minerals (which have zero molar volume anyway), and excludes any phase whose reported molar volume is between 499.9999 and 500.0001. The latter exclusion reflects the fact that the EQ3/6 databases currently assign a molar volume of 500.0000 cm^3 for phases whose molar volume is unknown. Note the oscillating, very small molar amounts for boron (B); these are rounding errors, and there is probably no real boron in the minerals. These small errors occur because the elemental composition of the minerals is obtained by starting with the EQ3/6 `mte` array, then subtracting out the contributions from fictive fixed fugacity "minerals", then subtracting out the aqueous phase array `mteaq`.

```
MOLES in minerals alone
data0.nuc.R8
|EQ6 input file name= F01_2022.6i
|Description= React J-13 water with metals in an FFTF waste package.
Years        zi           pH           VolMinerals(cc) O            Ag           Al           B
 3.7637E+03  3.7611E+02  6.0328E+00      7.0152E+02  9.2557E+01  1.0750E-02  1.8969E+00  0.0000E+00
 3.7637E+03  3.7611E+02  6.0328E+00      7.0152E+02  9.2557E+01  1.0750E-02  1.8969E+00  0.0000E+00
 3.7679E+03  3.7616E+02  5.8859E+00      7.0136E+02  9.2633E+01  1.0750E-02  1.8969E+00  5.0822E-21
 3.7883E+03  3.7642E+02  5.6844E+00      7.0054E+02  9.3001E+01  1.0750E-02  1.8969E+00  0.0000E+00
 3.8124E+03  3.7673E+02  5.5955E+00      6.9959E+02  9.3435E+01  1.0749E-02  1.8969E+00  2.5411E-20
 3.8137E+03  3.7674E+02  5.5919E+00      6.9959E+02  9.3445E+01  1.0749E-02  1.8969E+00  1.2096E-18
 3.8374E+03  3.7697E+02  5.5401E+00      6.9966E+02  9.3620E+01  1.0749E-02  1.8968E+00  6.7763E-21
```

A portion of the `elem_tot.txt` file is shown below. This file also contains (4+*N*) columns, but with several important differences. First, the aqueous component is included (though the fictive fixed fugacity "minerals" have been subtracted out). Second, the elemental contents of the remaining "special reactants" (but not the `DISPLACER` reactants) have been included in the

sum. The special reactants are not part of what EQ3/6 defines as the equilibrium system, but are typically used to represent package materials, such as steels, glass and fuel. Consequently, adding in the special reactants allows a check of the total amount of loss of fissile materials and neutron absorbers, as a function of time, and serves as an easy metric for criticality calculations. The `VolSolids(cc)` column gives an estimate of the volume of total solids in the system, including minerals and special reactants. For this estimate to be accurate, real values of `vreac` must be included for each solid special reactant specified in the input file. In this case, the input file (`F01_202.6i`) specified 0.0 for the `vreac`, so the calculated volume of solids is the same as the calculated volume of minerals from the previous example.

```
MOLES in minerals, aqu & special react
data0.nuc.R8
|EQ6 input file name= F01_2022.6i
|Description= React J-13 water with metals in an FFTF waste package.
Years        zi           pH           VolSolids(cc) O            Ag           Al           B
 3.7637E+03 3.7611E+02 6.0328E+00     7.0152E+02 1.4842E+02 1.0752E-02 1.8969E+00 1.2386E-05
 3.7637E+03 3.7611E+02 6.0328E+00     7.0152E+02 1.4842E+02 1.0752E-02 1.8969E+00 1.2386E-05
 3.7679E+03 3.7616E+02 5.8859E+00     7.0136E+02 1.4848E+02 1.0752E-02 1.8969E+00 1.2386E-05
 3.7883E+03 3.7642E+02 5.6844E+00     7.0054E+02 1.4878E+02 1.0752E-02 1.8969E+00 1.2386E-05
 3.8124E+03 3.7673E+02 5.5955E+00     6.9959E+02 1.4912E+02 1.0751E-02 1.8969E+00 1.2386E-05
 3.8137E+03 3.7674E+02 5.5919E+00     6.9959E+02 1.4914E+02 1.0751E-02 1.8969E+00 1.2386E-05
 3.8374E+03 3.7697E+02 5.5401E+00     6.9966E+02 1.4942E+02 1.0751E-02 1.8969E+00 1.2386E-05
```

The `elem_m_a.txt` file is similar to the three described above, except it contains elemental information for the sum of the minerals and aqueous phases.

5.0 RECOMMENDATIONS

5.1 Software Elements Validated

This SQR tested the addendum to EQ3/6 version 7.2b, which adds the capability to run the solid-centered flow-through mode in a single calculation (that is, with out the need to chain multiple EQ6 runs). The solid-centered flow-through mode was validated through the tests described in sections 3.1 and 3.2. In section 3.3, it was shown that the addition of this new mode did not alter the pre-existing functionality of EQ6, for the 17 "`.61`" test cases provided in the LLNL distribution of EQ3/6 version 7.2b. These 17 test cases exercise the B-Dot and Pitzer activity corrections, kinetic and non-kinetic modes, temperature jump calculations, titrations, and closed and open systems. The test cases do not exercise every conceivable combination of rate laws and solid solution models in EQ6, but the tests are quite comprehensive. No software limitations were discovered in the process of running the test cases (section 3), apart from the recommendation that the parameter `dlzmx2` be kept ≤ 0.2 when the `DISPLACER` mode is in use.

5.2 Software Elements Not Validated

All elements and components for running EQ6 reaction path calculations were validated.

5.3 Recommendations for Software Qualification

The EQ3/6 version 7.2b software is transferred to the M&O from LLNL. LLNL has performed an extensive validation of the EQ3/6 version 7.2 software, and has qualified the code for use in the Yucca Mountain Project. The changes described in this addendum do not affect the existing capabilities of EQ6, and have been shown to implement correctly the solid-centered flow-through mode. Therefore, it is recommended that this software addendum be qualified for use by the Waste Package Operations and Performance Assessment groups at the M&O to calculate the degradation of waste package (WPs) in water.

6.0 REFERENCES

1 Wolery, Thomas 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. Livermore, California: Lawrence Livermore National Laboratory. TIC Cat. No. 205087.

2 Daveler, Stephanie A., and Wolery, Thomas 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. Livermore, California: Lawrence Livermore National Laboratory. TIC Cat. No. 205240.

3 Wolery, Thomas 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. Livermore, California: Lawrence Livermore National Laboratory. TIC Cat. No. 205154.

4 Wolery, Thomas J., and Daveler, Stephanie 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. Livermore, California: Lawrence Livermore National Laboratory. TIC Cat. No. 205002.

5 Harrar, J. E., Carley, J. F., Isherwood, W. F., and Raber, E. 1990. *Report of the Committee to Review the Use of J-13 Well Water in Nevada Nuclear Waste Storage Investigations.* UCID-21867. Livermore, California: Lawrence Livermore National Laboratory. MOL.19980416.0660.

6 Civilian Radioactive Waste Management System (CRWMS) Management & Operating Contractor (M&O) 1998a. *Total System Performance Assessment-Viability Assessment (TSPA-VA) Analyses Technical Basis Document. Chapter 10, Disruptive Events.* B00000000-01717-4301-010 REV00. Las Vegas, Nevada: M&O. MOL.19980724.0399.

7 CRWMS M&O 1998b. *EQ6 Calculations for Chemical Degradation of PWR and MOX Spent Fuel Waste Packages.* BBA000000-01717-0210-00009 REV 00B. Las Vegas, Nevada: M&O. MOL.19980701.0483.

8 Stockman, H.W. 1994. *PP: A Graphics Post-Processor for the EQ6 Reaction Path Code.* SAND94-1955. Albuquerque, New Mexico: Sandia National Laboratories. TIC Cat. No. 241246.

9 CRWMS M&O 1998d. *EQ3/6 Software Installation and Testing Report for Pentium Based Personal Computers (PCs).* CSCI: LLYMP9602100. Las Vegas, Nevada: M&O. MOL.19980813.0191.

10 CRWMS M&O 1998e. *EQ6 Calculations for Chemical Degradation of Pu-Ceramic Waste Packages.* BBA000000-01717-0210-00018 REV 00. Las Vegas, Nevada: M&O. MOL.19980918.0004.

Attachment I: Source Code Changes

Changes to File eq6.for (to Create eq6new.for)

For each change listed below, several lines of the unchanged code are included to allow the programmer to find the proper position in the program and subroutines.

1. Header comments added:

```
c eq6
      program eq6
c****** Comment Header Block from H.W. Stockman *********
c 11-7-98 add hwsdis to display notice of alterations.
c 10-19-98 increase # special reactants from 10 to 12 for FFTF.
c 7-5-98 for M&O-type use, replace modr() with morr()
c 7-4-98 increase kpar from 90 to 110
c Clean re-build of source, from that
c obtained from T.J. Wolery (via SCM), to include these options:
c (1) Denote with hws## : output to create hwsdata file,
c     to be used by PP postprocessor.  Does not alter any
c     data or program flow in original EQ6 algorithms. A
c     new subroutine, hws4io, has been added, but this
c     routine merely outputs, and does not alter, extant
c     eq6 data.
c     Also include: write statements to print step
c     number, time (when in time mode), and minerals or
c     solid solutions in the system whenever an assemblage
c     changes. The purpose of these write statements is to
c     better inform the user about the status of the program,
c     to allow the user to terminate a stalled run or to
c     allow the user to estimate the time to completion.
c     Also: alter rtcalc to pass nhydr for hws4io use.
c     Also: call hwsdis to print M&O-specific disclaimer.
c=============================================================
c (2) Denote with plc : Changes suggested by Paul Cloke,
c     specifically to array sizes and minor write statements,
c     to allow for larger numbers of special reactants and
c     species.  These changes do not affect existing
c     algorithms or data flow, and are:
c-----------------------------------------------
c  In file eqlib.for, subroutine initim, the data statement for
c  the number of days in each month was corrected; as supplied
c  this statement showed 31 days in November and 30 for December.
c
c  Globally, in both the eq6.for and eqlib.for files, the
c  dimensions were increased for four arrays.  The dimensions are
c  changed though parameter statements; specifically the following
c  parameters were changed.
c  (A) The parameter nstpar (maximum number of aqueous species)
c    was increased from 800 to 2000.  This change also required a
c    change in format statement 2030 of subroutine echoz in
c    eq6.for, to permit proper printing of the changed array
c    dimension.
c  (B) The parameter nxmdpa (maximum number of species/reactions that
c    can be affected by the suppress/alter options) was increased
```

```
c     from 20 to 60.
c   (C) The parameter nrctpa (maximum number of specified reactant
c     species) was increased from 25 to 50.
c   (D) The parameter nsrtpa (maximum number of special reactant
c     species) was increased from 5 to 10. 10-19-98: 10 to 12.
c=============================================================
c (3) Denote with hws!! : changes for flow-through, rock-
c      centered mode.  This mode is signaled by jcode of 5.
c      These changes are active only when jcode 5 is specified.
c      The specific changes required for this mode are summarized
c      here:
c------ Begin summary specific changes, jcode 5 flow-through------
c Global change: add jhws(nrctpa) to eq6re.h, by replacing all
c  "jcode(nrctpa)," with "jcode(nrctpa),jhws(nrctpa),\n     $ "
c  where the '\n' represents a new line (carriage return)
c  inserted manually.
c Change to subroutine reacts: add block to elseif that starts
c  after label 53 and ends with line 43. Add eq6mta.h common so
c  we can reference mteaq.
c Changes to subroutine rd6rea: wherever "jcode(nrct) =" appears in
c  old code, add "jhws(nrct) = 0"; and add the block
c  starting "else if (uheader.eq.'DISPLACER'" to include case for
c  READ uvalue(2).eq.'5', which will reset jcode to 2 and set jhws
c  to 1...
c NOTES: the "displacer" type of reactant is indicated with
c  "DISPLACER" in Daveler format, and with a pseudo jcode of 5
c  in Wolery format.  Internally, the W format pseudo jcode of 5
c  is reset to 2, and the corresponding jhws flag is set to 1.
c  Thus this type of reactant is treated as a normal special
c  reactant everywhere except where jhws is tested -- currently
c  that is only in the subroutine reacts.
c NOTE: to get true continuum mixing, it is important to
c  adjust the dlzmx1 and dlzmx2 params in the input file
c  so that no step will be larger than about 20% of a "mole"
c  of the displacer reactant, and a "mole" of the displacer
c  reactant should comprise 1000 grams if 1000 grams of initial
c  solution are to be displaced.
c------ End summary specific changes, jcode 5 flow-through------
c****** END Comment Header Block from H.W. Stockman *****
c==== ORIGINAL YMP HEADER BELOW 6-9-98 =====
c
c     YMP Configuration Item Identifier: EQ3/6-V7-EQ6-SRC-R137
c     Last revised 10/11/95 by TJW
```

[...]

2. Globally, the jhws array is added to this header block:

```
c-----------------------------------------------------------------------
c eq6re.h
      common /eq6re/ mwtrc(nrctpa),vreac(nrctpa),morr(nrctpa),
     $ modr(nrctpa),moor(nrctpa),morrs(nrctpa),modrs(nrctpa),
     $ zircts(nrctpa),zirct(nrctpa),rxbar(iktpar,nxrtpa),
     $ rxbarb(iktpar,nxrtpa),cesrb(nctpar,nsrtpa),cesr(nctpar,nsrtpa),
     $ jreac(nrctpa),jcode(nrctpa),jhws(nrctpa),
     $ nrndex(nrctpa),nxridx(nxrtpa)
```

[...]

3. In program eq6 (main program), add hwsdis to print notice that this is a modified eq6:

```
      call prcndi(noutpt,nttyo)
c hws## 11-7-98 add notice of modification (1 line below).
      call hwsdis(noutpt,nttyo)
```

[...]

4. In program eq6 (main program), add call to rtcalc:

```
        call rsatch(qreq)
c hws## alter 1 line below: pass nhydr to rtcalc
        call rtcalc(nhydr)
      endif
      qftpr2=.false.
c
c     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
c
      call scripz(iopg2,nhydr,nchlor,qhydth,qftpr2)
```

[...]

5. In program eq6 (main program), add 2 lines:

```
c     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
c
c     call routine path to trace the reaction path.
c
      if (kstpmx.gt.0) call path(axmatr,xbarh,xqkx,csp,nsp,ntpr,
     $ nhydr,nchlor,itracx,iebal,nsqmx1,iapxmx,iktmax)

c hws## write w/ -1 only if there are steps...
      call hws4io(kstep, xi, xlkeh, nhydr)
      if(kstep.gt.1) call hws4io(-1, xi, xlkeh, nhydr)
c
c     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

[...]

6. Add entire subroutine, for producing binary output file (this function is not necessary for the solid-centered flow-through mode):

```
c hws## ====== entire subroutine ==========================
c hws4io  subroutine to output binary file called hwsdata for
c PP postprocessor.
c This routine is based on the code published in:
c Stockman, H.W. (1994) "PP: A Graphical Postprocessor for the
c   EQ6 Reaction Path Code", SAND94-1955, Sandia National Laboratories
c   (55 pp).
c-----------------------------------------------
      subroutine hws4io(kstep, xxi, xxlkeh, nnhydr)

      implicit logical(q),integer(i,j,k,n),real*8(a-h,l,m,o,p,r-t,v-z),
     $ character*8(u)
c-----------------------------------------------------------------------
      parameter(ntitpa=100,iktpar=20,imchpa=4,kpar=110)
      parameter(nctpar=100,ndctpa=4,nffgpa=15)
      parameter(ngtpar=80,nmtpar=850,nordpa=6)
      parameter(nprspa=50,nrctpa=50,nsqpar=200)
```

```
      parameter(nsrtpa=12,nstpar=2000,nxmdpa=60)
      parameter(nxoppa=10,nxpepa=15,nxrtpa=12)
      parameter(nxtpar=50,iapxpa=12,narxpa=5,ntprpa=2)
c
      parameter(nctpa1=nctpar+1,nsqpa1=nsqpar+1,nordp1=nordpa+1)
      parameter(nrstpa=nstpar-nctpar-1)
      parameter(nsspar=nxtpar*iktpar)
c-----------------------------------------------------------------------
c eq6af.h
      common /eq6af/ aff(nmtpar),affx(iktpar,nxtpar),afflcx(nxtpar),
     $ newsat,ksat,kxsat
c-----------------------------------------------------------------------
c eq6dd.h
      common /eq6dd/ azero(nstpar),glg(nstpar),glgo(nstpar),
     $ conc(nstpar),conclg(nstpar),act(nstpar),actlg(nstpar),
     $ hydn(nstpar)
c-----------------------------------------------------------------------
c eq6dzi.h    zi1 is needed here, don't pass zi
      common /eq6dzi/ zi1,zi2,zi3,tstrt,time,dltime,times,timemx,
     $ tdays,tyears,tlogs,tlogd,tlogyr,qriinf
c-----------------------------------------------------------------------
c eq6ff.h
      common /eq6ff/ xilg,fo2lg,xic,dshmc,shmc,fo2
c-----------------------------------------------------------------------
c eq6gg.h
      character*24 uspec,umin,ugas,usolx
      common /eq6gg/ uspec(nstpar),umin(nmtpar),ugas(ngtpar),
     $ usolx(nxtpar),uelem(nctpa1),uoxide(nctpar)
c-----------------------------------------------------------------------
c eq6hh.h
c
c       tzero = last temp for which tstep was called
c       told = temp (c) at previous point of reaction prograess
c       tempc = temp (c)
c       tempk = temp (k)
c       press = pressure, bars
c       tempc = temperature (c)
c       tempc0 = temp (c) at zi1 = 0 or time = 0
c       tk1, tk2, tk3 = temperature tracking coefficients (see
c                       routine tstep)
c       jtemp = temperature tracking option parameter (see routine
c               tstep)
c       tdamax = data file nominal upper temperature limit
c       tdamin = data file nominal lower temperature limit
c
      common /eq6hh/ tzero,told,tempc,tempc0,tempc1,tempc2,tempk,press,
     $ tk1,tk2,tk3,tdamax,tdamin,jtemp
c-----------------------------------------------------------------------
c eq6ka.h
      common /eq6ka/ iindx1(kpar),jsflag(nstpar),jmflag(nmtpar),
     $ jkflag(iktpar,nxtpar),jxflag(nxtpar),jgflag(ngtpar),
     $ jcflag(nctpar),jflag(nsqpar),ibasis(nstpar),ibswx(nsqpar),
     $ qbswx
c-----------------------------------------------------------------------
c eq6lm.h 7-5-98 hws## note old 7.2a had lomx(iktpar,nstpar) bug!
      common /eq6lm/ moss(nstpar),loss(nstpar),momn(nmtpar),
     $ lomn(nmtpar),momx(iktpar,nxtpar),lomx(iktpar,nxtpar),
```

```
     $ motx(nxtpar),lotx(nxtpar),mosst,actlw,xbarlw,xbarw,
     $ actlwc,xbrlwc,xbarwc
c-----------------------------------------------------------------------
c eq6mt.h
      common /eq6mt/ mte(nctpar),mtes(nctpar),electr,qnochb
c-----------------------------------------------------------------------
c eq6mta.h
      common /eq6mta/ mteaq(nctpar),mteaqb(nctpar),fuglg(ngtpar),
     $ fug(ngtpar)
c-----------------------------------------------------------------------
c eq6ni.h
      common /eq6ni/ nct,nsb,nsb1,nsq,nsq1,nst,nrst,nmt,nxt,ngt,nctd
c-----------------------------------------------------------------------

c eq6qq.h
      common /eq6qq/ qbad,qflag,qdump
c-----------------------------------------------------------------------
c eq6rc.h
      common /eq6rc/ nrct,nrctmx,nsrt,nsrtmx,nart,nmrt,nxrt,nxrtmx,
     $ ngrt
c-----------------------------------------------------------------------
c eq6re.h
      common /eq6re/ mwtrc(nrctpa),vreac(nrctpa),morr(nrctpa),
     $ modr(nrctpa),moor(nrctpa),morrs(nrctpa),modrs(nrctpa),
     $ zircts(nrctpa),zirct(nrctpa),rxbar(iktpar,nxrtpa),
     $ rxbarb(iktpar,nxrtpa),cesrb(nctpar,nsrtpa),cesr(nctpar,nsrtpa),
     $ jreac(nrctpa),jcode(nrctpa),jhws(nrctpa),
     $ nrndex(nrctpa),nxridx(nxrtpa)
c
      character*24 ureac,uendb
      common /eq6rec/ ureac(nrctpa),uendb(iktpar,nxrtpa),
     $  uesrb(nctpar,nsrtpa)
c-----------------------------------------------------------------------
c eqlpp.h
      common /eqlpp/ all0,farad,rconst,afcnst,ehfac,volg,om,omlg,
     $ omi,tcnst1,tcnst2
c-----------------------------------------------------------------
c eq6ki.h
      common /eq6ki/ kct,ksb,ksq,km1,kmt,kx1,kxt,kdim,kmax
c-----------------------------------------------------------------------
c eq6si.h
      common /eq6si/ jct,jst,jrst,jmt,jxt,jgt
c-----------------------------------------------------------------
c eq6ti.h
      common /eq6ti/ ntitl1,ntitl2,ntitld,ntitmx
c
      character*80 utitl1,utitl2,utitld
      common /eq6tic/ utitl1(ntitpa),utitl2(ntitpa),utitld(ntitpa)
c-----------------------------------------------------------------
      dimension ihws(nmtpar)
      dimension ihit(nmtpar)
c     !below 2 statements for solid solution data
      dimension ixhws(nxtpar)
      dimension ixhit(nxtpar)
c     !affmul is 1/afcnst; 2-27-94 add ixplac, etc for sol soln
      save ihws,iplace,ixhws,ixplac,nlu,nerr,ictr,affmul
      save ihit, ixhit
```

```
      data ictr/0/, ihit/nmtpar*0/, ixhit/nxtpar*0/

      if(kstep.eq.1) then
           affmul = 1./afcnst
c          print*, char(7), '++++++++++++++++++++++++++'
           iplace = 1
           ixplac = 1
c          !if the file exists, destroy it
           inquire(file='hwsdata',exist=qex)
           if(qex) then
                call getlu(nlu,nerr)
                open(unit=nlu,status='old',file='hwsdata')
                close(unit=nlu,status='delete')
           endif
c          !find an open device
           call getlu(nlu,nerr)
c          !BEGIN_MACHINE_DEPENDENT_CODE_LAHEY
           open(unit=nlu, access='transparent', status='unknown',
     &          file='hwsdata')
c          !END_MACHINE_DEPENDENT_CODE_LAHEY
c          !*******Header; logicals implicitly 4bytes*******
c          !write database name size, then name
           write(nlu) len(utitld(1)), utitld(1)
c          !write a descriptive line from input file;
c          !If 'D' format, use second line...
c           if(utitl1(1)(1:1).eq.'|') then
c                write(nlu) len(utitl1(2)), utitl1(2)
c           else
                write(nlu) len(utitl1(1)), utitl1(1)
c           endif
c          !END writing descriptive header
           write(nlu) qdump, qnochb
c          !write number of active...
           write(nlu) kct, jgt, jmt, jst, jxt
c          !Master var names ph,eh,fo2,xi,zi,time,tempc, press
c          ! not written; assume these in C code...
c          !Write the # reactants and the reactant names
           write(nlu) nrct
           do 20 i=1,nrct
                if(jcode(i).eq.1) then
                     write(nlu) usolx(nrndex(i))
                else
                     write(nlu) ureac(i)
                endif
   20      continue
c          !Write names element,species,minerals for log Q/K, gases
c- - - - - - - - - - - - - -
           do 21 i=1,nct
                if(jcflag(i).gt.0) write(nlu) uelem(i)
   21      continue
c- - - - - - - - - - - - - -
           do 22 i=1,nst
                if(jsflag(i).le.0) write(nlu) uspec(i)
   22      continue
c- - - - - - - - - - - - - -
           do 23 i=1,nmt
                if(jmflag(i).le.0) write(nlu) umin(i)
```

```
   23       continue
c- - - - - - - - - - - - - - -
c iictr
            iictr = 0
            do 24 i=1,nxt
                 if(jxflag(i).le.0) then
                      if(iictr.lt.jxt) write(nlu) usolx(i)
                      iictr = iictr + 1
                 endif
   24       continue
c- - - - - - - - - - - - - - -
            do 25 i=1,ngt
                 if(jgflag(i).le.0) write(nlu) ugas(i)
   25       continue
c- - - - - - - - - - - - - - -
      endif
c--------END of HEADER BLOCK------------------------
c A kstep.eq.-1 signals it is time to write the trailer and close
c the file.  In the future, we could close the file with every
c step, which would allow a user to check the status of
c a long run at any point; but we would have to back up and delete
c the footer.
      if(kstep.eq.-1) then
c           !append the min names and number of minerals
            if(iplace.gt.1) write(nlu) (umin(ihws(i)), i=1,iplace-1)
c           !Last 12 bytes are the # min, # sol soln, and # plot points
c           !Add distinctive tag at end, to check run ended properly
            iiitag = 123456789
            if(ixplac.gt.1) write(nlu) (usolx(ixhws(i)), i=1,ixplac-1)
            write(nlu) iplace - 1, ixplac - 1, ictr, iiitag
            close(nlu)
      else
c           !!! time display
            if(time.gt.0.0) then
                 write(*,*) 'step=',kstep,'  log10(seconds)=',tlogs
            else
                 write(*,*) 'step=',kstep
            endif
c Next line (commented out) useful for debugging
c           write(*,*) '==>nmt,nxt,iplace,ixplac',nmt,nxt,iplace,ixplac
c           !Write data for this step
            ictr = ictr + 1
            !calculate the ph, eh
            ph = -actlg(nnhydr)
            if(.not.qnochb) then
                 eh = ehfac*0.25*(fo2lg - 4.*ph - 2.*actlg(1) - xxlkeh)
            else
                 eh = -999.
            endif
c           !write out the "master" variables; add tempc, press...
            write(nlu) ph,eh,fo2,xxi,zi1,time,tempc,press
c           !write out amounts of remaining reactants
c           !changed from destroyed reactants 7-5-98 for M&O-type use
            write(nlu) (morr(i), i=1,nrct)
c- - - - - - - - - - - - - - -
            do 31 i=1,nct
                 if(jcflag(i).gt.0) write(nlu) mteaq(i)
```

```
   31       continue
c- - - - - - - - - - - - - -
            do 32 i=1,nst
                 if(jsflag(i).le.0) write(nlu) conc(i)
   32       continue
c- - - - - - - - - - - - - - the log (Q/K) below
            do 33 i=1,nmt
                 if(jmflag(i).le.0) write(nlu) aff(i)*affmul
   33       continue
c- - - - - - - - - - - - - -
            do 35 i=1,ngt
                 if(jgflag(i).le.0) write(nlu) fug(i)
   35       continue
c- - - - - - - - - - - - - -
c the ihws and ixhws arrays track the existence of pure
c minerals and solid solutions, respectively; we don't write
c data for minerals unless they are in the system.
c           !update ihws array and write current step data
            do 100 i=1,nmt
                 if((jmflag(i).eq.-1).and.(ihit(i).eq.0)) then
                      ihit(i) = 1
                      ihws(iplace) = i
                      iplace = iplace + 1
                 endif
  100       continue
c           !write this step's mineral amount data
            write(nlu) iplace-1
            if(iplace.gt.1) write(nlu) (momn(ihws(i)), i=1,iplace-1)
c- - - - - - - - - - similar for solid sol'ns
c           !update ixhws array and write current step data
            do 200 i=1,nxt
                 if((jxflag(i).eq.-1).and.(ixhit(i).eq.0)) then
                      ixhit(i) = 1
                      ixhws(ixplac) = i
                      ixplac = ixplac + 1
                 endif
  200       continue
c           !write this step's sol soln amount data
            write(nlu) ixplac-1
            if(ixplac.gt.1) write(nlu) (motx(ixhws(i)), i=1,ixplac-1)
       endif
       end
c=== END hws## subroutine hws4io ========================
```

[...]

7. In subroutine echoz :

```
c plc change format below to allow printing 4 digits (i4)
      write (noutpt,2030) nct,nctmax,kct,nstmax,nst,jst,nrstmx,nrst,jrst
 2030 format(/4x,'No. of elements in the data base = ',i3,/
     $ 4x,'No. of elements dimensioned for = ',i3,/
     $ 4x,'No. of active elements = ',i3,//
     $ 7x,'No. of aqueous species dimensioned for = ',i4,/
     $ 7x,'No. of aqueous species loaded = ',i4,/
     $ 7x,'No. of active aqueous species = ',i4,//
     $ 10x,'No. of aqueous reactions dimensioned for = ',i4,/
     $ 10x,'No. of aqueous reactions loaded = ',i4,/
```

```
     $ 10x,'No. of active aqueous reactions = ',i4,/)
      write (noutpt,2031) nmtmax,nmt,jmt,ngtmax,ngt,jgt,nxt,nxtmax,jxt
```

[...]

8. In subroutine modexz, several minor changes for informative screen output:

```
c-----------------------------------------------------------------------
c
c hws## 1 line below: make it more obvious where we are
      write(*,*) '==========CHANGING PHASE ASSEMBLAGE=========='
      kcol=ksq
      do 15 nm=1,nmt
      if (jmflag(nm).eq.-1 .or. jmflag(nm).eq.-10) then
        kcol=kcol+1
        if (kcol.le.kmax) then
          iindx1(kcol)=nm
          uzvec1(kcol)=umin(nm)
          jmflag(nm)=-1
          lxx=lomn(nm)
          zvclg1(kcol)=lxx
          zvec1(kcol)=texp(lxx)
c hws## 1 line below, to show min. assem.
          write(*,*) '----------', umin(nm), '----------'
        else
          write (noutpt,1005) umin(nm)
          write (nttyo,1005) umin(nm)
 1005     format(/' * error - (eq6/modexz) Have exceeded the Jacobian',
     $    /7x,'matrix dimension (kpar) trying to put "',a24,'"',
     $    /7x,'in the equilibrium system.')
        endif
      endif
   15 continue
      kmt=kcol
      km1=ksq+1
      if (nxt.gt.0. and. iopt4.gt.0) then
        do 25 nx=1,nxt
c hws## 1 line below: add k1hws, solid sol'n indicator
        k1hws = 0
        ncpt=ncomp(nx)
        do 20 ik=1,ncpt
        if (jkflag(ik,nx).eq.-1 .or. jkflag(ik,nx).eq.-10) then
          kcol=kcol+1
          if (kcol.le.kmax) then
            iindx1(kcol)=50000+100*nx+ik
            uzvec1(kcol)(1:8)=usolx(nx)(1:8)
            nm=nend(ik,nx)
            if (nm .gt. 0) then
              uzvec1(kcol)(9:24)=umin(nm)(1:16)
              jkflag(ik,nx)=-1
              lxx=lomx(ik,nx)
              zvclg1(kcol)=lxx
              zvec1(kcol)=texp(lxx)
            else
              kcol=kcol-1
            endif
c hws## 1 line added below: mark solid sol'n
            k1hws = 1
```

```
          else
            write (noutpt,1005) usolx(nx)
            write (nttyo,1005) usolx(nx)
            endif
          endif
   20     continue
c hws## 1 line below: when changing solid sol'n
          if(k1hws.eq.1) write(*,*)':::::::::::', usolx(nx), ':::::::::::'
   25     continue
       endif
       kdim=kcol
       kx1=kmt+1
       kxt=kcol
c
       if (kdim.gt.kmax) then
         write (noutpt,1015) kdim,kmax
         write (nttyo,1015) kdim,kmax
 1015    format(/9x,'Needed the matrix dimension (kpar) to be at least ',
     $   i5,'.')
         stop
       endif
c
       end
```

[...]

9. In subroutine path, add several calls to output-only functions:

```
c-----------------------------------------------------------------
c
c      save information at the current point, compute new finite
c      differences, and estimate derivatives at the latest point
c      of reaction progress.
c
   20 kstep=kstep+1
c hws## next line added
       call hws4io(kstep, xi, xlkeh, nhydr)
       kstppr=kstppr+1
       kstppl=kstppl+1
       i=kstep-1
```

[...]

```
          write (nttyo,2006) kstpze
 2006     format(/' * error - (eq6/path) The step size (delzi) has',
     $    /7x,'been zero ',i3,' times in a row. Stopping execution.')
c hws## next 3 lines; incl. 1 comment:
c write w/ -1 only if there are steps...
          call hws4io(kstep, xi, xlkeh, nhydr)
          if(kstep.gt.1) call hws4io(-1, xi, xlkeh, nhydr)
          stop
        endif
      else
```

[...]

```
 2007     format(/' * error - (eq6/path) The step size (delzi) has',
     $    /7x,'not exceeded the minimum value ',i3,' times in a row.',
     $    /7x,'Stopping execution.')
c hws## next 3 lines, incl 1 comment:
c         write w/ -1 only if there are steps...
```

```
          call hws4io(kstep, xi, xlkeh, nhydr)
          if(kstep.gt.1) call hws4io(-1, xi, xlkeh, nhydr)
          stop
        endif
      else
        kstpmn = 0
      endif
[...]
 2017   format(/" * error - (eq6/path) Can't recover by going back",
     $  /7x,'to the last point of reaction progress at which',
     $  /7x,"calculations were successful. Can't write a pickup file",
     $  /7x,'describing the system at that point. Try re-running the',
     $  /7x,'problem, setting the maximum number of steps (kstpmx)',
     $  /7x,'to the value which corresponds to that point. Then',
     $  /7x,'try restarting the calculation from that point.',
     $  /7x,'Stopping execution.')
c hws## next 3 lines, incl 1 comment:
c       write w/ -1 only if there are steps...
        call hws4io(kstep, xi, xlkeh, nhydr)
        if(kstep.gt.1) call hws4io(-1, xi, xlkeh, nhydr)
        stop
      endif
[...]
c     test for sufficient accuracy in integrating the rate equations
c
c hws## alter 1 line below to pass nhydr
      call rtcalc(nhydr)
      if (iopt1.le.0) go to 1298
      if (qriinf) then
        if (delzi.le.dlzimn) then
          write (noutpt,1272)
          write (nttyo,1272)
 1272     format(/' * note - (eq6/path) Have made an infinite time ',
[...]
```

10. In subroutine rd6rea, add:

```
            call upcase(uvalue(2), uheadr)
            if (uheadr.eq.'MINERAL'.or.uvalue(2).eq.'0') then
               jcode(nrct) = 0
c hws!! add 1 line below:
               jhws(nrct) = 0
               nmrt = nmrt + 1
            else if (uheadr.eq.'SOLID SOLUTION'.or.
     $         uvalue(2).eq.'1') then
               jcode(nrct) = 1
c hws!! add 1 line below:
               jhws(nrct) = 0
               if (nxrt .ge. nxrtpa) then
                  ierrct = ierrct + 1
                  write(noutpt, 15) nxrtpa, linect
                  write(nttyo,  15) nxrtpa, linect
               else
                  nxrt = nxrt + 1
               endif
            else if (uheadr.eq.'SPECIAL'.or.uvalue(2).eq.'2') then
```

```
                jcode(nrct) = 2
c hws!! add 1 line below
                jhws(nrct) = 0
                if (nsrt .ge. nsrtpa) then
                    ierrct = ierrct + 1
                    write(noutpt, 15) nsrtpa, linect
                    write(nttyo,  15) nsrtpa, linect
                else
                    nsrt = nsrt + 1
                endif
c hws!! BEGIN new block for displacer-type reactant
             else if (uheadr.eq.'DISPLACER'.or.uvalue(2).eq.'5') then
                jcode(nrct) = 2
                jhws(nrct) = 1
                if (nsrt .ge. nsrtpa) then
                    ierrct = ierrct + 1
                    write(noutpt, 15) nsrtpa, linect
                    write(nttyo,  15) nsrtpa, linect
                else
                    nsrt = nsrt + 1
                endif
c hws!! END new block for displacer-type reactant
             else if (uheadr.eq.'AQUEOUS'.or.uvalue(2).eq.'3') then
                jcode(nrct) = 3
c hws!! add 1 line below:
                jhws(nrct) = 0
                nart = nart + 1
             else if (uheadr.eq.'GAS'.or.uvalue(2).eq.'4') then
                jcode(nrct) = 4
c hws!! add 1 line below:
                jhws(nrct) = 0
                ngrt = ngrt + 1
             else
                ierrct = ierrct + 1
                write(noutpt, 13) uvalue(2), linect
                write(nttyo,  13) uvalue(2), linect
             endif
```

[...]

11. In subroutine `reacts`, add `eq6mta.h` block so one can reference `mteaq`:

```
c hws!! add eq6mta.h block so can ref. mteaq .
c-----------------------------------------------------------------------
c eq6mta.h
      common /eq6mta/ mteaq(nctpar),mteaqb(nctpar),fuglg(ngtpar),
     $ fug(ngtpar)
c-----------------------------------------------------------------
c
c eq6bl.h
      common cdrs(nsqpal,nrstpa),cess(nctpar,nstpar),cdrst(nrstpa),
     $ store(nstpar)
c-----------------------------------------------------------------------
```

[...]

12. In subroutine `reacts`, make the change that implements solid-centered flow-through:

```
c hws!! comment out next 4 lines old code
```

```
c       elseif (jcode(nrc).eq.2) then
c         do 43 nc=1,nct
c         mte(nc)=mte(nc)+dmodr*cesr(nc,nr)
c   43    continue
c hws!! BEGIN block substitution of above old code
      elseif (jcode(nrc).eq.2) then
        if(jhws(nrc).eq.0) then
          do 43 nc=1,nct
          mte(nc)=mte(nc)+dmodr*cesr(nc,nr)
   43     continue
        else
          do 430 nc=1,nct
          mte(nc)=mte(nc)+dmodr*(cesr(nc,nr) - mteaq(nc))
  430     continue
        endif
c hws!! END block substitution
```

[...]

13. In subroutine `readz`:

```
 2049 format(3x,'reactant= ',a24,/6x,'jcode= ',i2,15x,'jreac= ',i2,/
     $ 7x,'morr= ',1pe12.5,6x,'modr= ',e12.5)
c hws!! old code: jcode(nrc)=jdum
c hws!! BEGIN new code block
      if(jdum.eq.5) then
        jdum = 2
        jhws(nrc) = 1
      else
        jhws(nrc) = 0
      endif
      jcode(nrc) = jdum
c hws!! END new code block
```

[...]

14. Alter subroutine `rtcalc` to pass nhydr and to close binary file if program aborts:

```
c hws## alter rtcalc to pass nhydr
c rtcalc
      subroutine rtcalc(nhydr)
c
c   this routine calculates the relative and (for iopt1.ge.1) absolute
```

[...]

```
  310     format(' ---  actions to be taken-   check input ',
     $    'file for negative rate constants,',/29x,
     $    'set switch iodb9=2 in the input',//)
c hws## next 3 lines, incl 1 comment:
c         write w/ -1 only if there are steps...
          call hws4io(kstep, xi, xlkeh, nhydr)
          if(kstep.gt.1) call hws4io(-1, xi, xlkeh, nhydr)
          stop
        end if
c
        if ( (affr.lt.0. .and. .not.qovstp) .and.
     $        rrelr1(nrc).gt.0. ) then
          write (noutpt,290) ureac(nrc),rreac1(nrc),rrelr1(nrc)
          write (noutpt,320)
```

```
  320     format(' * error- calculated rate is .gt. 0, but',
     $    ' affinity is .lt. 0  ---',/'        (reactant is ',
     $    'dissolving, but it should precipitate) (rtcalc)')
          write (noutpt,310)
c hws## next 3 lines, incl 1 comment:
c         write w/ -1 only if there are steps...
          call hws4io(kstep, xi, xlkeh, nhydr)
          if(kstep.gt.1) call hws4io(-1, xi, xlkeh, nhydr)
          stop
        end if
      endif
```

[...]

15. Add simple routine hwsdis to warn user this is a modified program:

```
c*************************************************************
c hws## 11-7-98 simple routine to print a warning that this
c  is an altered version of eq6 for M&O use.

      subroutine hwsdis(noutpt,nttyo)
c
c     This routine is called by eq6.f immediately after prcndi and
c     after the copyright notice ending with "All rights reserved."
c     The present routine writes an additional paragraph concerning
c     modifications made for the M&O work, and the version.
c
c
c-----------------------------------------------------------------------

      write (noutpt,1000)
      write (nttyo,1000)
 1000 format(' -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-',
$ /' THIS IS EQ6NEW.FOR, AN ALTERED VERSION OF EQ6 V7.2B, ',
     $ /' R137.  The alterations consist of a flow-through mode',
     $ /' (jcode=5 or displacer reactant) and I/O functions to ',
     $ /' display progress and to produce binary output files. ',
     $ /' Contact HWStockman (Harlan_Stockman@notes.ymp.gov)   ',
     $ /' for more information. EQ6NEW version 1.0, 11-07-98.  ',
     $ /' MEDIA # 30084-M04-001 for this Addendum only.        ',
     $ /' CRWMS M&O; original document # UCRL-MA-110662.       ',
     $ /' -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-',//)
end
```

16. Add subroutine hwstxt to create the elem*.txt files. This routine affects only output; since it is lengthy and similar to hws4io (see change 6), the reader is referred to the electronic distribution for source code.

Changes to File eqlib.for (to Create eqlibnew.for)

1. In subroutine initim:

```
c       data ndays(1) /31/
c       data (ndays(mo), mo = 3,12) /31,30,31,30,31,31,30,31,30,31/
c     plc, 2/13/98.  Discovered that the number of days for Nov. and
```

```
c     Dec. were reversed.  Corrected.
[...]

c* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
c
c     BEGIN_PC_DEPENDENT_CODE
com
c hws## adds mon to line below for pp
         integer mo,mm,dd,yy,hr,mi,se,hs,mon
        integer ndays(12)
[...]
        if (mod(yy,4) .eq. 0) ndays(2) = 29
        iexec0 = dd
c hws## change mo to mon 2 and 3 lines below
        if (mm .gt. 1) then
          do mon = 1, mm - 1
            iexec0 = iexec0 + ndays(mon)
          enddo
        endif
[...]
```

2. In subroutine runtim:

```
c     BEGIN_PC_DEPENDENT_CODE
com
c hws## adds mon below
        integer mo,mm,dd,yy,iy,hr,mi,se,hs,idays,iexec1,jexec1,mon
        integer ndays(12)
[...]
        if (mm .gt. 1) then
c hws## change mo to mon, both of two lines below
          do mon = 1, mm - 1
            iexec1 = iexec1 + ndays(mon)
          enddo
        endif
        jexec1 = yy
[...]
```

3. In subroutine timdat:

```
c       udate(1:2) = utd(9:10)
c       udate(3:5) = utd(5:7)
c       udate(6:7) = utd(23:24)
c       udate(8:8) = ublank
com
c     plc, 2/13/98.  Appended (1:8) to "utime"
c       utime(1:8) = utd(12:19)
com
c     END_UNIX_DEPENDENT_CODE
c
[...]
```

Attachment II: Test Case Checklist

EQ6NEW FLOW-THROUGH UPDATE FOR EQ3/6 PACKAGE

Pentium PC Test Case Checklist for CPU#_____
Date_____

This installation test procedure requires that a previous installation of EQ3/6 exists on the system, per the instructions in Ref. 9. It is also assumed that the supplement has been installed and the `eq6new.exe` binary has been copied to `eq6.exe` (in the directory indicated by the `EQ36CO` environment variable in the `autoexec.bat` file), as described in sections 2.2 and 2.3.1.

If a "NO GO" condition is encountered, contact Software Configuration Management.

1. Verify that directory `eq6new` contains five supplemental test case files:

	GO	NOGO
`_micro.6i`	___	___
`_heatqf.6i`	___	___
`_pptqtza.6i`	___	___
`_fwbrmix.6i`	___	___
`new_01.6i`	___	___

The first four of these are identical in content to the `micro.6i`, `heatqf.6i`, `pptqtza.6i`, and `fwbrmix.6i` files in the LLNL EQ3/6 7.2b distribution, except the names have been pre-pended with the underscore character to avoid confusion. The last file (`new_01.6i`) is specifically designed to test the new solid-centered flow-through mode.

2. Verify that directory `eq6new` also contains reference output from the five supplemental test case files:

	GO	NOGO
`_micro.6nw`	___	___
`_heatqf.6nw`	___	___
`_pptqtza.6nw`	___	___
`_fwbrmix.6nw`	___	___
`new_01.6nw`	___	___

These outputs were generated with `eq6new.exe`, compiled as described in section 2.2. (The ending `.6nw` is not a standard EQ6 file extension, and is used solely to prevent the reference files from being overwritten during testing.)

3. Run the test cases by entering the `eq6new` directory and typing at the DOS prompt:

	GO	NOGO
`runeq6 com _micro.6i`	___	___
`runeq6 com _heatqf.6i`	___	___
`runeq6 com _pptqtza.6i`	___	___
`runeq6 hmw _fwbrmix.6i`	___	___
`runeq6 com new_01.6i`	___	___

Note that `runeq6` renames the files, created by `eq6`, based on the input file name. In the case of the input file `_micro.6i`, the output file is `_micro.6o`, the pickup file is `_micro.6p`, the tab file is `_micro.6t`, and the tabx file is `_micro.6tx`. Note also that the `_fwbrmix` case must be run with the hmw database.

4. Verify that the following output files now exist in the `eq6new` directory:

	GO	NOGO
`_micro.6o`	___	___
`_heatqf.6o`	___	___
`_pptqtza.6o`	___	___
`_fwbrmix.6o`	___	___
`new_01.6o`	___	___

5. Obtain and write down the revision numbers for `EQ6` and `EQLIB`, for both the "`.6nw`" and the "`.6o`" files generated in the `eq6new` directory. The revision numbers will be found in text lines at the top of the output files; the lines will resemble (example for `.6o`):

```
EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R136-P5)
Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)
```

In the example above, `R136-P5` is the revision number for `EQ6`, and `R168-P5` is the revision number for `EQLIB`.

Write the revision numbers for the installation below:

	.6nw files in eq6new:	.6o files in eq6new:
EQ6:	_____	_____
EQLIB:	_____	_____

6. Use the DOS `fc` command to compare the reference output files (`.6nw` files) and the `.6o` files in directory `eq6new`. Enter the `eq6new` directory, then use the DOS `fc` command, redirecting the output to `*.txt` files. Type the following at the DOS prompt:

	GO	NOGO
`cd \eq6new`	___	___
`fc _micro.6nw _micro.6o > _micro.txt`	___	___
`fc _heatqf.6nw _heatqf.6o > _heatqf.txt`	___	___
`fc _pptqtza.6nw _pptqtza.6o > _pptqtza.txt`	___	___
`fc _fwbrmix.6nw _fwbrmix.6o > _fwbrmix.txt`	___	___
`fc new_01.6nw new_01.6o > new_01.txt`	___	___

7. Examine each output from the `fc` command (`txt` file) to verify that the differences between the reference (`.6nw`) and test output (`.6o`) files are small. Typically the only differences between the two file sets will be in the dates. Because of slight changes in the implementation of transcendental functions in successive generations of Intel processors, there may also be small differences in the last significant digits of the steps sizes and convergence measures.

	GO	NOGO
`_micro.txt`	___	___
`_heatqf.txt`	___	___
`_pptqtza.txt`	___	___
`_fwbrmix.txt`	___	___
`new_01.txt`	___	___

8. Sign below that the verification test has been completed.

_____ Test Performer

_____ Date

Attachment III: Concatenated fc (File Compare) Outputs for Section 3.3 Tests

```
fc DEDOLO.6o   DEDOLO.6nw   > dedolo.txt
fc HEATQF.6o   HEATQF.6nw   > heatqf.txt
fc J13WSF.6o   J13WSF.6nw   > J13WSF.txt
fc J13WTUFF.6o J13WTUFF.6nw > J13WTUFF.txt
fc METHANE.6o  METHANE.6nw  > METHANE.txt
fc MICRO.6o    MICRO.6nw    > MICRO.txt
fc MICROFT.6o  MICROFT.6nw  > MICROFT.txt
fc PPTCALA.6o  PPTCALA.6nw  > PPTCALA.txt
fc PPTQTZ.6o   PPTQTZ.6nw   > PPTQTZ.txt
fc PPTQTZA.6o  PPTQTZA.6nw  > PPTQTZA.txt
fc RWTITR.6o   RWTITR.6nw   > RWTITR.txt
fc EVSWGYHA.6o EVSWGYHA.6nw > EVSWGYHA.txt
fc CALHAL.6o   CALHAL.6nw   > CALHAL.txt
fc FWBRMIX.6o  FWBRMIX.6nw  > FWBRMIX.txt
fc GYPNACL.6o  GYPNACL.6nw  > GYPNACL.txt
fc GYPSUMB.6o  GYPSUMB.6nw  > GYPSUMB.txt
fc CRISQTZ.6o  CRISQTZ.6nw  > CRISQTZ.txt
Comparing files dedolo.6o and DEDOLO.6nw
****** dedolo.6o
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-P5)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)

****** DEDOLO.6nw
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-Pentium)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-Pentium)

******

****** dedolo.6o

 Run  14:06:41  12/11/98

****** DEDOLO.6nw

 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-MO4-001 for this Addendum only.
 CRWMS M&O; original document # UCRL-MA-110662.
 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

******

****** dedolo.6o
```

```
****** DEDOLO.6nw

 Run  13:58:07  12/11/98



******

****** dedolo.6o

       No. of aqueous species dimensioned for = 800
       No. of aqueous species loaded = 280
       No. of active aqueous species =  30

          No. of aqueous reactions dimensioned for = 699
          No. of aqueous reactions loaded = 200
          No. of active aqueous reactions =  22

****** DEDOLO.6nw

       No. of aqueous species dimensioned for = 2000
       No. of aqueous species loaded =  280
       No. of active aqueous species =   30

          No. of aqueous reactions dimensioned for = 1899
          No. of aqueous reactions loaded =  200
          No. of active aqueous reactions =   22

******

****** dedolo.6o

          Start time = 14:06:41  12/11/98
            End time = 14:06:45  12/11/98

           Run time =   4.06     seconds
          User time =   4.06     seconds
           Cpu time =   4.06     seconds

****** DEDOLO.6nw

          Start time = 13:58:07  12/11/98
            End time = 13:58:11  12/11/98

           Run time =   4.01     seconds
          User time =   4.01     seconds
           Cpu time =   4.01     seconds

******


Comparing files heatqf.6o and HEATQF.6nw
****** heatqf.6o
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-P5)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)
```

```
****** HEATQF.6nw
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-Pentium)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-Pentium)

******

****** heatqf.6o

 Run  14:06:48  12/11/98

****** HEATQF.6nw

 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-MO4-001 for this Addendum only.
 CRWMS M&O; original document # UCRL-MA-110662.
 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

******

****** heatqf.6o

****** HEATQF.6nw

 Run  13:58:13  12/11/98


******

****** heatqf.6o

       No. of aqueous species dimensioned for = 800
       No. of aqueous species loaded = 351
       No. of active aqueous species = 229

          No. of aqueous reactions dimensioned for = 699
          No. of aqueous reactions loaded = 271
          No. of active aqueous reactions = 216

****** HEATQF.6nw

       No. of aqueous species dimensioned for = 2000
       No. of aqueous species loaded =  351
       No. of active aqueous species =  229

          No. of aqueous reactions dimensioned for = 1899
          No. of aqueous reactions loaded =  271
          No. of active aqueous reactions =  216
```

```
******

****** heatqf.6o

          Start time = 14:06:48  12/11/98
            End time = 14:06:54  12/11/98

           Run time =    5.55     seconds
          User time =    5.55     seconds
           Cpu time =    5.55     seconds

****** HEATQF.6nw

          Start time = 13:58:13  12/11/98
            End time = 13:58:18  12/11/98

           Run time =    4.66     seconds
          User time =    4.66     seconds
           Cpu time =    4.66     seconds

******


Comparing files j13wsf.6o and J13WSF.6nw
****** j13wsf.6o
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-P5)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)

****** J13WSF.6nw
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-Pentium)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-Pentium)

******

****** j13wsf.6o

 Run  14:06:55  12/11/98

****** J13WSF.6nw

 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-MO4-001 for this Addendum only.
 CRWMS M&O; original document # UCRL-MA-110662.
 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

******

****** j13wsf.6o
```

```
****** J13WSF.6nw

 Run  13:58:19  12/11/98


******

****** j13wsf.6o

       No. of aqueous species dimensioned for = 800
       No. of aqueous species loaded = 721
       No. of active aqueous species = 649

          No. of aqueous reactions dimensioned for = 699
          No. of aqueous reactions loaded = 641
          No. of active aqueous reactions = 616

****** J13WSF.6nw

       No. of aqueous species dimensioned for = 2000
       No. of aqueous species loaded =  721
       No. of active aqueous species =  649

          No. of aqueous reactions dimensioned for = 1899
          No. of aqueous reactions loaded =  641
          No. of active aqueous reactions =  616

******

****** j13wsf.6o

          Start time = 14:06:55  12/11/98
            End time = 14:07:43  12/11/98

           Run time =   47.5     seconds
          User time =   47.5     seconds
           Cpu time =   47.5     seconds

****** J13WSF.6nw

          Start time = 13:58:19  12/11/98
            End time = 13:59:02  12/11/98

           Run time =   43.2     seconds
          User time =   43.2     seconds
           Cpu time =   43.2     seconds

******


Comparing files j13wtuff.6o and J13WTUFF.6nw
****** j13wtuff.6o
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-P5)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)
```

```
****** J13WTUFF.6nw
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-Pentium)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-Pentium)

******

****** j13wtuff.6o

 Run  14:07:43  12/11/98

****** J13WTUFF.6nw

 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-MO4-001 for this Addendum only.
 CRWMS M&O; original document # UCRL-MA-110662.
 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

******

****** j13wtuff.6o


****** J13WTUFF.6nw

 Run  13:59:03  12/11/98



******

****** j13wtuff.6o

       No. of aqueous species dimensioned for = 800
       No. of aqueous species loaded = 459
       No. of active aqueous species = 346

          No. of aqueous reactions dimensioned for = 699
          No. of aqueous reactions loaded = 379
          No. of active aqueous reactions = 327

****** J13WTUFF.6nw

       No. of aqueous species dimensioned for = 2000
       No. of aqueous species loaded =  459
       No. of active aqueous species =  346

          No. of aqueous reactions dimensioned for = 1899
          No. of aqueous reactions loaded =  379
          No. of active aqueous reactions =  327
```

```
******

****** j13wtuff.6o

          Start time = 14:07:43  12/11/98
            End time = 14:08:11  12/11/98

           Run time =   27.9     seconds
          User time =   27.9     seconds
           Cpu time =   27.9     seconds

****** J13WTUFF.6nw

          Start time = 13:59:03  12/11/98
            End time = 13:59:32  12/11/98

           Run time =   28.8     seconds
          User time =   28.8     seconds
           Cpu time =   28.8     seconds

******


Comparing files methane.6o and METHANE.6nw
****** methane.6o
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-P5)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)

****** METHANE.6nw
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-Pentium)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-Pentium)

******

****** methane.6o

 Run  14:08:13  12/11/98

****** METHANE.6nw

 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-MO4-001 for this Addendum only.
 CRWMS M&O; original document # UCRL-MA-110662.
 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

******

****** methane.6o
```

```
|-----------------------------------------------------------------------|
****** METHANE.6nw

 Run  13:59:33  12/11/98


|-----------------------------------------------------------------------|
******

****** methane.6o

       No. of aqueous species dimensioned for = 800
       No. of aqueous species loaded = 331
       No. of active aqueous species = 201

          No. of aqueous reactions dimensioned for = 699
          No. of aqueous reactions loaded = 251
          No. of active aqueous reactions = 192

****** METHANE.6nw

       No. of aqueous species dimensioned for = 2000
       No. of aqueous species loaded =  331
       No. of active aqueous species =  201

          No. of aqueous reactions dimensioned for = 1899
          No. of aqueous reactions loaded =  251
          No. of active aqueous reactions =  192

******

****** methane.6o

          Start time = 14:08:13  12/11/98
            End time = 14:08:25  12/11/98

           Run time =   12.4     seconds
          User time =   12.4     seconds
           Cpu time =   12.4     seconds

****** METHANE.6nw

          Start time = 13:59:33  12/11/98
            End time = 13:59:46  12/11/98

           Run time =   13.1     seconds
          User time =   13.1     seconds
           Cpu time =   13.1     seconds

******


Comparing files micro.6o and MICRO.6nw
****** micro.6o
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-P5)
```

```
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)

****** MICRO.6nw
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-Pentium)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-Pentium)

******

****** micro.6o

 Run  14:08:27  12/11/98

****** MICRO.6nw

 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-M04-001 for this Addendum only.
 CRWMS M&O; original document # UCRL-MA-110662.
 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

******

****** micro.6o

|-----------------------------------------------------------------------|
****** MICRO.6nw

 Run  13:59:48  12/11/98


|-----------------------------------------------------------------------|
******

****** micro.6o

       No. of aqueous species dimensioned for = 800
       No. of aqueous species loaded = 196
       No. of active aqueous species =  21

          No. of aqueous reactions dimensioned for = 699
          No. of aqueous reactions loaded = 116
          No. of active aqueous reactions =  15

****** MICRO.6nw

       No. of aqueous species dimensioned for = 2000
       No. of aqueous species loaded =  196
       No. of active aqueous species =   21

          No. of aqueous reactions dimensioned for = 1899
          No. of aqueous reactions loaded =  116
```

```
          No. of active aqueous reactions =   15

******

****** micro.6o

          Start time = 14:08:27  12/11/98
            End time = 14:08:28  12/11/98

           Run time =   1.21     seconds
          User time =   1.21     seconds
           Cpu time =   1.21     seconds

****** MICRO.6nw

          Start time = 13:59:48  12/11/98
            End time = 13:59:49  12/11/98

           Run time =   1.43     seconds
          User time =   1.43     seconds
           Cpu time =   1.43     seconds

******


Comparing files microft.6o and MICROFT.6nw
****** microft.6o
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-P5)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)

****** MICROFT.6nw
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-Pentium)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-Pentium)

******

****** microft.6o

 Run  14:08:29  12/11/98

****** MICROFT.6nw

 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-MO4-001 for this Addendum only.
 CRWMS M&O; original document # UCRL-MA-110662.
 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

******
```

```
****** microft.6o

|-----------------------------------------------------------------------|
****** MICROFT.6nw

 Run  13:59:50  12/11/98


|-----------------------------------------------------------------------|
******

****** microft.6o

       No. of aqueous species dimensioned for = 800
       No. of aqueous species loaded = 196
       No. of active aqueous species =  21

          No. of aqueous reactions dimensioned for = 699
          No. of aqueous reactions loaded = 116
          No. of active aqueous reactions =  15

****** MICROFT.6nw

       No. of aqueous species dimensioned for = 2000
       No. of aqueous species loaded =  196
       No. of active aqueous species =   21

          No. of aqueous reactions dimensioned for = 1899
          No. of aqueous reactions loaded =  116
          No. of active aqueous reactions =   15

******

****** microft.6o

          Start time = 14:08:29  12/11/98
            End time = 14:08:31  12/11/98

           Run time =   1.59     seconds
          User time =   1.59     seconds
           Cpu time =   1.59     seconds

****** MICROFT.6nw

          Start time = 13:59:50  12/11/98
            End time = 13:59:52  12/11/98

           Run time =   1.76     seconds
          User time =   1.76     seconds
           Cpu time =   1.76     seconds

******


Comparing files pptcala.6o and PPTCALA.6nw
****** pptcala.6o
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
```

```
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-P5)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)

****** PPTCALA.6nw
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-Pentium)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-Pentium)

******

****** pptcala.6o

 Run  14:08:32  12/11/98

****** PPTCALA.6nw

 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-M04-001 for this Addendum only.
 CRWMS M&O; original document # UCRL-MA-110662.
 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

******

****** pptcala.6o

|-----------------------------------------------------------------------|
****** PPTCALA.6nw

 Run  13:59:53  12/11/98


|-----------------------------------------------------------------------|
******

****** pptcala.6o

      No. of aqueous species dimensioned for = 800
      No. of aqueous species loaded = 250
      No. of active aqueous species =  10

         No. of aqueous reactions dimensioned for = 699
         No. of aqueous reactions loaded = 170
         No. of active aqueous reactions =   6

****** PPTCALA.6nw

      No. of aqueous species dimensioned for = 2000
      No. of aqueous species loaded =  250
      No. of active aqueous species =   10

         No. of aqueous reactions dimensioned for = 1899
```

```
          No. of aqueous reactions loaded =  170
          No. of active aqueous reactions =    6

******

****** pptcala.6o

          Start time = 14:08:32  12/11/98
            End time = 14:08:33  12/11/98

           Run time =  0.940     seconds
          User time =  0.940     seconds
           Cpu time =  0.940     seconds

****** PPTCALA.6nw

          Start time = 13:59:53  12/11/98
            End time = 13:59:54  12/11/98

           Run time =   1.10     seconds
          User time =   1.10     seconds
           Cpu time =   1.10     seconds

******


Comparing files pptqtz.6o and PPTQTZ.6nw
****** pptqtz.6o
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-P5)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)

****** PPTQTZ.6nw
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-Pentium)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-Pentium)

******

****** pptqtz.6o

 Run  14:08:34  12/11/98

****** PPTQTZ.6nw

 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-MO4-001 for this Addendum only.
 CRWMS M&O; original document # UCRL-MA-110662.
 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

******
```

```
****** pptqtz.6o

|-----------------------------------------------------------------------|
****** PPTQTZ.6nw

 Run  13:59:55  12/11/98


|-----------------------------------------------------------------------|
******

****** pptqtz.6o

       No. of aqueous species dimensioned for = 800
       No. of aqueous species loaded = 248
       No. of active aqueous species =  11

          No. of aqueous reactions dimensioned for = 699
          No. of aqueous reactions loaded = 168
          No. of active aqueous reactions =   7

****** PPTQTZ.6nw

       No. of aqueous species dimensioned for = 2000
       No. of aqueous species loaded =  248
       No. of active aqueous species =   11

          No. of aqueous reactions dimensioned for = 1899
          No. of aqueous reactions loaded =  168
          No. of active aqueous reactions =    7

******

****** pptqtz.6o

          Start time = 14:08:34  12/11/98
            End time = 14:08:34  12/11/98

           Run time =  0.440     seconds
          User time =  0.440     seconds
           Cpu time =  0.440     seconds

****** PPTQTZ.6nw

          Start time = 13:59:55  12/11/98
            End time = 13:59:56  12/11/98

           Run time =  0.600     seconds
          User time =  0.600     seconds
           Cpu time =  0.600     seconds

******


Comparing files pptqtza.6o and PPTQTZA.6nw
****** pptqtza.6o
```

```
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-P5)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)

****** PPTQTZA.6nw
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-Pentium)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-Pentium)

******

****** pptqtza.6o

 Run  14:08:36  12/11/98

****** PPTQTZA.6nw

 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-M04-001 for this Addendum only.
 CRWMS M&O; original document # UCRL-MA-110662.
 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

******

****** pptqtza.6o

|-----------------------------------------------------------------------|
****** PPTQTZA.6nw

 Run  13:59:57  12/11/98


|-----------------------------------------------------------------------|
******

****** pptqtza.6o

       No. of aqueous species dimensioned for = 800
       No. of aqueous species loaded = 248
       No. of active aqueous species =  11

          No. of aqueous reactions dimensioned for = 699
          No. of aqueous reactions loaded = 168
          No. of active aqueous reactions =   7

****** PPTQTZA.6nw

       No. of aqueous species dimensioned for = 2000
       No. of aqueous species loaded =  248
       No. of active aqueous species =   11
```

```
          No. of aqueous reactions dimensioned for = 1899
          No. of aqueous reactions loaded =  168
          No. of active aqueous reactions =    7

******

****** pptqtza.6o

          Start time = 14:08:36  12/11/98
            End time = 14:08:37  12/11/98

           Run time =   1.53     seconds
          User time =   1.53     seconds
           Cpu time =   1.53     seconds

****** PPTQTZA.6nw

          Start time = 13:59:57  12/11/98
            End time = 13:59:59  12/11/98

           Run time =   1.81     seconds
          User time =   1.81     seconds
           Cpu time =   1.81     seconds

******


Comparing files rwtitr.6o and RWTITR.6nw
****** rwtitr.6o
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-P5)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)

****** RWTITR.6nw
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-Pentium)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-Pentium)

******

****** rwtitr.6o

 Run  14:08:39  12/11/98

****** RWTITR.6nw

 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-MO4-001 for this Addendum only.
 CRWMS M&O; original document # UCRL-MA-110662.
 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

```
******

****** rwtitr.6o

|-----------------------------------------------------------------------|
****** RWTITR.6nw

 Run  14:00:01  12/11/98

|-----------------------------------------------------------------------|
******

****** rwtitr.6o

       No. of aqueous species dimensioned for = 800
       No. of aqueous species loaded = 422
       No. of active aqueous species = 304

          No. of aqueous reactions dimensioned for = 699
          No. of aqueous reactions loaded = 342
          No. of active aqueous reactions = 288

****** RWTITR.6nw

       No. of aqueous species dimensioned for = 2000
       No. of aqueous species loaded =  422
       No. of active aqueous species =  304

          No. of aqueous reactions dimensioned for = 1899
          No. of aqueous reactions loaded =  342
          No. of active aqueous reactions =  288

******

****** rwtitr.6o

          Start time = 14:08:39  12/11/98
            End time = 14:08:50  12/11/98

           Run time =   11.6     seconds
          User time =   11.6     seconds
           Cpu time =   11.6     seconds

****** RWTITR.6nw

          Start time = 14:00:01  12/11/98
            End time = 14:00:12  12/11/98

           Run time =   11.0     seconds
          User time =   11.0     seconds
           Cpu time =   11.0     seconds

******


Comparing files evswgyha.6o and EVSWGYHA.6nw
```

```
****** evswgyha.6o
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-P5)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)

****** EVSWGYHA.6nw
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-Pentium)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-Pentium)

******

****** evswgyha.6o

 Run  14:09:11  12/11/98

****** EVSWGYHA.6nw

 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-MO4-001 for this Addendum only.
 CRWMS M&O; original document # UCRL-MA-110662.
 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

******

****** evswgyha.6o

****** EVSWGYHA.6nw

 Run  14:04:49  12/11/98


******

****** evswgyha.6o

       No. of aqueous species dimensioned for = 800
       No. of aqueous species loaded =  17
       No. of active aqueous species =  16

          No. of aqueous reactions dimensioned for = 699
          No. of aqueous reactions loaded =   7
          No. of active aqueous reactions =   7

****** EVSWGYHA.6nw

       No. of aqueous species dimensioned for = 2000
       No. of aqueous species loaded =   17
       No. of active aqueous species =   16
```

```
          No. of aqueous reactions dimensioned for = 1899
          No. of aqueous reactions loaded =    7
          No. of active aqueous reactions =    7

******

****** evswgyha.6o

          Start time = 14:09:11  12/11/98
            End time = 14:09:12  12/11/98

           Run time =  0.830     seconds
          User time =  0.830     seconds
           Cpu time =  0.830     seconds

****** EVSWGYHA.6nw

          Start time = 14:04:49  12/11/98
            End time = 14:04:50  12/11/98

           Run time =   1.10     seconds
          User time =   1.10     seconds
           Cpu time =   1.10     seconds

******


Comparing files calhal.6o and CALHAL.6nw
****** calhal.6o
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-P5)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)

****** CALHAL.6nw
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-Pentium)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-Pentium)

******

****** calhal.6o

 Run  14:09:09  12/11/98

****** CALHAL.6nw

 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-MO4-001 for this Addendum only.
 CRWMS M&O; original document # UCRL-MA-110662.
 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

```
******

****** calhal.6o

****** CALHAL.6nw

 Run  14:04:47  12/11/98


******

****** calhal.6o

       No. of aqueous species dimensioned for = 800
       No. of aqueous species loaded =  14
       No. of active aqueous species =  10

          No. of aqueous reactions dimensioned for = 699
          No. of aqueous reactions loaded =   4
          No. of active aqueous reactions =   4

****** CALHAL.6nw

       No. of aqueous species dimensioned for = 2000
       No. of aqueous species loaded =   14
       No. of active aqueous species =   10

          No. of aqueous reactions dimensioned for = 1899
          No. of aqueous reactions loaded =    4
          No. of active aqueous reactions =    4

******

****** calhal.6o

          Start time = 14:09:09  12/11/98
            End time = 14:09:10  12/11/98

           Run time =  0.390     seconds
          User time =  0.390     seconds
           Cpu time =  0.390     seconds

****** CALHAL.6nw

          Start time = 14:04:47  12/11/98
            End time = 14:04:48  12/11/98

           Run time =  0.500     seconds
          User time =  0.500     seconds
           Cpu time =  0.500     seconds

******
```

```
Comparing files fwbrmix.6o and FWBRMIX.6nw
****** fwbrmix.6o
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-P5)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)

****** FWBRMIX.6nw
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-Pentium)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-Pentium)

******

****** fwbrmix.6o

 Run  14:09:13  12/11/98

****** FWBRMIX.6nw

 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-M04-001 for this Addendum only.
 CRWMS M&O; original document # UCRL-MA-110662.
 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

******

****** fwbrmix.6o


****** FWBRMIX.6nw

 Run  14:04:51  12/11/98



******

****** fwbrmix.6o

       No. of aqueous species dimensioned for = 800
       No. of aqueous species loaded =  14
       No. of active aqueous species =  10

          No. of aqueous reactions dimensioned for = 699
          No. of aqueous reactions loaded =   4
          No. of active aqueous reactions =   4

****** FWBRMIX.6nw

       No. of aqueous species dimensioned for = 2000
       No. of aqueous species loaded =   14
```

```
      No. of active aqueous species =   10

         No. of aqueous reactions dimensioned for = 1899
         No. of aqueous reactions loaded =    4
         No. of active aqueous reactions =    4

******

****** fwbrmix.6o

         Start time = 14:09:13  12/11/98
           End time = 14:09:13  12/11/98

          Run time =  0.220     seconds
         User time =  0.220     seconds
          Cpu time =  0.220     seconds

****** FWBRMIX.6nw

         Start time = 14:04:51  12/11/98
           End time = 14:04:52  12/11/98

          Run time =  0.380     seconds
         User time =  0.380     seconds
          Cpu time =  0.380     seconds

******


Comparing files gypnacl.6o and GYPNACL.6nw
****** gypnacl.6o
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-P5)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)

****** GYPNACL.6nw
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-Pentium)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-Pentium)

******

****** gypnacl.6o

 Run  14:09:14  12/11/98

****** GYPNACL.6nw

 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-MO4-001 for this Addendum only.
 CRWMS M&O; original document # UCRL-MA-110662.
```

```
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

******

****** gypnacl.6o

****** GYPNACL.6nw

 Run  14:04:53  12/11/98


******

****** gypnacl.6o

       No. of aqueous species dimensioned for = 800
       No. of aqueous species loaded =  12
       No. of active aqueous species =   8

          No. of aqueous reactions dimensioned for = 699
          No. of aqueous reactions loaded =   2
          No. of active aqueous reactions =   2

****** GYPNACL.6nw

       No. of aqueous species dimensioned for = 2000
       No. of aqueous species loaded =    12
       No. of active aqueous species =     8

          No. of aqueous reactions dimensioned for = 1899
          No. of aqueous reactions loaded =    2
          No. of active aqueous reactions =    2

******

****** gypnacl.6o

          Start time = 14:09:14  12/11/98
            End time = 14:09:14  12/11/98

           Run time =  0.380     seconds
          User time =  0.380     seconds
           Cpu time =  0.380     seconds

****** GYPNACL.6nw

          Start time = 14:04:53  12/11/98
            End time = 14:04:54  12/11/98

           Run time =  0.770     seconds
          User time =  0.770     seconds
           Cpu time =  0.770     seconds

******
```

```
Comparing files gypsumb.6o and GYPSUMB.6nw
****** gypsumb.6o
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-P5)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)

****** GYPSUMB.6nw
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-Pentium)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-Pentium)

******

****** gypsumb.6o

 Run  14:09:15  12/11/98

****** GYPSUMB.6nw

 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-M04-001 for this Addendum only.
 CRWMS M&O; original document # UCRL-MA-110662.
 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

******

****** gypsumb.6o

****** GYPSUMB.6nw

 Run  14:04:55  12/11/98


******

****** gypsumb.6o

       No. of aqueous species dimensioned for = 800
       No. of aqueous species loaded =  12
       No. of active aqueous species =   8

          No. of aqueous reactions dimensioned for = 699
          No. of aqueous reactions loaded =   2
          No. of active aqueous reactions =   2

****** GYPSUMB.6nw

       No. of aqueous species dimensioned for = 2000
```

```
      No. of aqueous species loaded =   12
      No. of active aqueous species =    8

         No. of aqueous reactions dimensioned for = 1899
         No. of aqueous reactions loaded =    2
         No. of active aqueous reactions =    2

******

****** gypsumb.6o

         Start time = 14:09:15  12/11/98
           End time = 14:09:15  12/11/98

          Run time =  0.170     seconds
         User time =  0.170     seconds
          Cpu time =  0.170     seconds

****** GYPSUMB.6nw

         Start time = 14:04:55  12/11/98
           End time = 14:04:55  12/11/98

          Run time =  0.330     seconds
         User time =  0.330     seconds
          Cpu time =  0.330     seconds

******


Comparing files crisqtz.6o and CRISQTZ.6nw
****** crisqtz.6o
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-P5)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-P5)

****** CRISQTZ.6nw
 EQ3/6, Version 7.2b (EQ3/6-V7-REL-V7.2b-PC)
 EQ6 Reaction-Path Code (EQ3/6-V7-EQ6-EXE-R137-Pentium)
 Supported by the EQLIB library (EQ3/6-V7-EQLIB-LIB-R168-Pentium)

******

****** crisqtz.6o

 Run  14:06:36  12/11/98

****** CRISQTZ.6nw

 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
 THIS IS EQ6NEW 1.0, AN ALTERED VERSION OF EQ6 V7.2B,
 R137.  The alterations consist of a flow-through mode
 (jcode=5 or displacer reactant) and I/O functions to
 display progress and to produce binary output files.
 Contact HWStockman (Harlan_Stockman@ymp.gov)
 for more information. EQ6NEW version 12-11-98.
 MEDIA # 30084-M04-001 for this Addendum only.
```

```
 CRWMS M&O; original document # UCRL-MA-110662.
 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

******

****** crisqtz.6o

****** CRISQTZ.6nw

 Run  13:58:02  12/11/98


******

****** crisqtz.6o

        No. of aqueous species dimensioned for = 800
        No. of aqueous species loaded = 184
        No. of active aqueous species =   8

           No. of aqueous reactions dimensioned for = 699
           No. of aqueous reactions loaded = 104
           No. of active aqueous reactions =   5

****** CRISQTZ.6nw

        No. of aqueous species dimensioned for = 2000
        No. of aqueous species loaded =  184
        No. of active aqueous species =    8

           No. of aqueous reactions dimensioned for = 1899
           No. of aqueous reactions loaded =  104
           No. of active aqueous reactions =    5

******

****** crisqtz.6o

           Start time = 14:06:36  12/11/98
             End time = 14:06:40  12/11/98

            Run time =   3.68     seconds
           User time =   3.68     seconds
            Cpu time =   3.68     seconds

****** CRISQTZ.6nw

           Start time = 13:58:02  12/11/98
             End time = 13:58:06  12/11/98

            Run time =   4.28     seconds
           User time =   4.28     seconds
            Cpu time =   4.28     seconds
```