

1/2

# SOFTWARE RELEASE NOTICE

1. SRN Number: <b>PA-SRN-216, PA-SRN-221, PA-SRN-222</b>		
2. Project Title: <b>Near-Field Environment KTI</b>		Project No. <b>01402.562,752, 561</b>
3. SRN Title: <b>MULTIFLO V.1.2, V.1.2.1, V.1.2.2</b>		
4. Originator/Requestor: <b>Bruce Mabrito</b>		Date: <b>1/19/2001</b>
5. Summary of Actions  <input type="checkbox"/> Release of new software  <input type="checkbox"/> Release of modified software: <input type="checkbox"/> Enhancements made <input type="checkbox"/> Corrections made  <input type="checkbox"/> Change of access software  <input checked="" type="checkbox"/> Software Retirement: <b>MULTIFLO V.1.2, V.1.2.1, V.1.2.2</b>		
6. Persons Authorized Access		
Name	Read Only/Read-Write	Addition/Change/Delete
<b>Scott Painter</b>	<b>RW</b>	<b>Addition/Change/Delete</b>
<b>Debra Hughson</b>	<b>RO</b>	
<b>Lauren Browning</b>	<b>RO</b>	
<b>Ron Green</b>	<b>RO</b>	
<b>Goodluck Ofoegbu</b>	<b>RO</b>	
<b>Melissa Hill</b>	<b>RO</b>	
<b>Walter Illman</b>	<b>RO</b>	
<b>Mohan Seth</b>	<b>RW</b>	
<b>English Percy</b>	<b>RO</b>	
<b>Bret Leslie</b>	<b>RO</b>	
<b>Hans Arlt</b>	<b>RO</b>	<b>Addition/Change/Delete</b>
7. Element Manager Approval: <i>E. C. Pey</i>		Date: <i>1/23/2001</i>
8. Remarks:		

**MULTIFLO**  
**V.1.2**

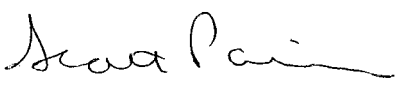
1/25

# SOFTWARE RELEASE NOTICE

1. SRN Number: <b>PA-SRN-216</b>		
2. Project Title: <b>Near-Field Environment KTI</b>		Project No. <b>20-1402-752</b> <sup>562</sup> <i>7/24/2000</i>
3. SRN Title: <b>MULTIFLO Version 1.2</b>		
4. Originator/Requestor: <b>Bruce Mabrito</b>		Date: <b>3/6/00</b>
5. Summary of Actions  <input type="checkbox"/> Release of new software  <input checked="" type="checkbox"/> Release of modified software:  <input checked="" type="checkbox"/> Enhancements made: DCM, (Unsturctured grid.)  <input checked="" type="checkbox"/> Corrections made: Bugs corrected.  <input type="checkbox"/> Change of access software  <input type="checkbox"/> Software Retirement		
6. Persons Authorized Access		
Name	Read Only/Read-Write	Addition/Change/Delete
Scott Painter	RW	Addition/Change/Delete
Debra Hughson	RO	
Lauren Browning	RO	
Ron Green	RO	
Goodluck Ofoegbu	RO	
Melissa Hill	RO	
Walter Illman	RO	Addition/Change/Delete
Mohan Seth	RW	
English Percy	RO	
Bret Leslie	RO	
7. Element Manager Approval: <i>[Signature]</i>		Date: <b>3/13/2000</b>
8. Remarks:		

2/25

## SOFTWARE SUMMARY FORM

01. Summary Date: <b>03/06/00</b>	02. Summary prepared by (Name and phone) <b>Scott Painter, 522-3348</b>	03. Summary Action:  <b>New</b>	
04. Software Date: <b>03/06/00</b>	05. Short Title: <b>MULTIFLO Version 1.2</b>		
06. Software Title:  <b>MULTIFLO Version 1.2</b>		07. Internal Software ID:  <b>NONE</b>	
08. Software Type:  <input type="checkbox"/> Automated Data System <input checked="" type="checkbox"/> Computer Program <input type="checkbox"/> Subroutine/Module	09. Processing Mode:  <input type="checkbox"/> Interactive <input type="checkbox"/> Batch <input checked="" type="checkbox"/> Combination	10. APPLICATION AREA a. General: <input checked="" type="checkbox"/> Scientific/Engineering <input checked="" type="checkbox"/> Auxiliary Analyses <input type="checkbox"/> Total System PA <input type="checkbox"/> Subsystem PA <input type="checkbox"/> Other  b. Specific: <b>Groundwater multiphase flow and reactive transport model</b>	
11. Submitting Organization and Address: <b>CNWR 6220 Culebra Road San Antonio, TX 78228</b>		12. Technical Contact(s) and Phone:  <b>Scott Painter, (210) 522-3348 Mohan Seth, (972) 699-3610</b>	
13. Narrative: <b>The code is used to model multiphase groundwater flow and reactive transport.</b>			
14. Computer Platform  <b>SUN</b>	15. Computer Operating System:  <b>UNIX</b>	16. Programming Language(s):  <b>Fortran 77</b>	17. Number of Source Program Statements:  <b>~80,000</b>
18. Computer Memory Requirements:  <b>Problem Dependent</b>	19. Tape Drives:  <b>N/A</b>	20. Disk/Drum Units:  <b>N/A</b>	21. Graphics:  <b>ASCII plot data files</b>
22. Other Operational Requirements <b>Thermodynamic database required.</b>			
23. Software Availability:  <input checked="" type="checkbox"/> Available <input type="checkbox"/> Limited <input type="checkbox"/> In-House ONLY		24. Documentation Availability:  <input checked="" type="checkbox"/> Available <input type="checkbox"/> Inadequate <input type="checkbox"/> In-House ONLY <b>DRAFT</b>	
Software Developer:  3-6-00 Date:			

**SOFTWARE  
REQUIREMENTS  
DESCRIPTION**

delivered  
3/12/88

IM 561-820

6/25

**SOFTWARE REQUIREMENTS DESCRIPTION  
FOR THE COMPUTER CODE MULTIFLO  
VERSION 1.2**

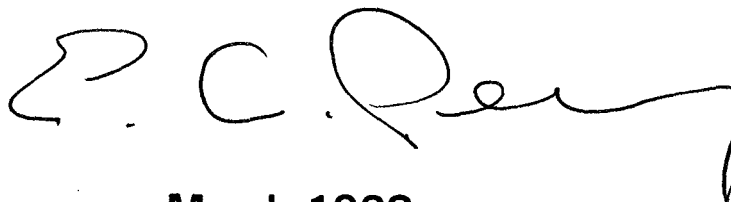
*Prepared for*

**Nuclear Regulatory Commission  
Contract NRC-02-97-009**

*Prepared by*

**Peter C. Lichtner**

**Center for Nuclear Waste Regulatory Analyses  
San Antonio, Texas**

A handwritten signature in black ink, appearing to read 'P. C. Lichtner', with a long horizontal flourish extending to the right.

**March 1998**

7/25

## ABSTRACT

This Software Requirements Description presents proposed revisions to the computer code MULTIFLO to implement a dual continuum model (DCM) capability. The DCM will be used in place of the equivalent continuum model in thermal-hydrological-chemical modeling of the proposed high-level nuclear waste repository at Yucca Mountain, Nevada.

# CONTENTS

Section	Page
TABLE .....	iv
ACKNOWLEDGMENTS .....	v
QUALITY OF DATA, ANALYSES, AND CODE DEVELOPMENT .....	v
1 INTRODUCTION .....	1
2 SOFTWARE REQUIREMENT DESCRIPTION MULTIFLO, VERSION 1.2 .....	1
2.1 SOFTWARE FUNCTION .....	1
2.2 BACKGROUND .....	1
2.3 CURRENT STATUS OF MULTIFLO .....	2
2.4 DUAL CONTINUUM MODEL IMPLEMENTATION IN MULTIFLO .....	2
2.5 RELATED CODES .....	3
3 IMPLEMENTATION .....	3
4 REFERENCES .....	4



## TABLES

Table	Page
3-1 Estimation of dual continuum model implementation time . . . . .	3

## ACKNOWLEDGMENTS

This report was prepared to document work performed by the Center for Nuclear Waste Regulatory Analyses (CNWRA) for the Nuclear Regulatory Commission (NRC) under Contract No. NRC-02-97-009. The activities reported here were performed on behalf of the NRC Office of Nuclear Material Safety and Safeguards (NMSS), Division of Waste Management (DWM). The report is an independent product of the CNWRA and does not necessarily reflect the views or regulatory position of the NRC.

## QUALITY OF DATA, ANALYSES, AND CODE DEVELOPMENT

**DATA:** CNWRA-generated original data contained in this report meets quality assurance requirements described in the CNWRA Quality Assurance Manual. Sources for other data should be consulted for determining the level of quality for those data.

**ANALYSES AND CODES:** No analyses work is reported in this document. This document describes planned changes to software.

# 1 INTRODUCTION

This Software Requirements Description (SRD) document describes proposed revision to the computer code MULTIFLO, a numerical model describing multiphase, multicomponent, reactive transport in a variably saturated porous medium. This software could be used in the high-level waste (HLW) repository license application review process for Yucca Mountain (YM).

The code can be used to address the very-near-field (drift scale), and near-field (repository scale) performance of the repository. The code can be applied to such processes as:

- (i) isothermal and nonisothermal liquid and vapor phase movement of water through unsaturated rock at YM.
- (ii) predicting the evolution of groundwater compositions near and within the engineered barrier system.
- (iii) predicting changes in porosity and permeability of the host rock resulting from mineral alteration and their effect on fluid transport.
- (iv) prediction of transport of aqueous and gaseous radionuclides from the waste package.

## 2 SOFTWARE REQUIREMENT DESCRIPTION: MULTIFLO, VERSION 1.2

This SRD briefly outlines the software function, technical basis, and computational approach, that are relevant to the proposed enhancements of the code MULTIFLO. Version 1.0 of MULTIFLO has been completed and satisfies TOP-018 QA requirements. A SRD was completed for Version 1.0 of MULTIFLO. A draft version of the User's Manual has been completed. The revised code will be issued as Version 1.2.

### 2.1 SOFTWARE FUNCTION

Planned change to the code MULTIFLO is to include a dual continuum model (DCM) capability. The DCM will be used both to replace and enhance the multiple interacting continua (MINC) approach for modeling highly fractured porous rock at the YM proposed repository site. The MINC method is currently being programmed into both METRA and GEM modules which will be released as Version 2.0 of MULTIFLO. The programming language used in MULTIFLO is FORTRAN. The code will be developed on a Sun-sparc workstation and PCs running NEXTSTEP and PC-UNIX.

### 2.2 BACKGROUND

The main purpose of this revision to MULTIFLO to incorporate the DCM is to aid in providing a detailed model of the near-field environment from which total performance assessment analyses may be abstracted. Both Lawrence Livermore National Laboratory (LLNL) and Lawrence Berkeley National Laboratory (LBNL) have begun using the DCM almost exclusively in place of the equivalent continuum model (ECM). The ECM is based on the assumption of capillary equilibrium between matrix and

fractures which is much too stringent to explain field observations of pore water chemistry at YM, including recent observations of  $^{36}\text{Cl}$  and differences in matrix and fracture solution chemistry. Observations of  $^{36}\text{Cl}$  at the proposed repository horizon indicate the existence of fast pathways from the ground surface to the watertable which are presumed related to flow through fractures. To describe such situations it is important to be able to distinguish between fracture and matrix flow systems.

Two available alternative approaches to the ECM, one the DCM and the other the MINC model (Pruess and Narisimhan, 1980), have been applied to YM. The DCM is applicable to the case where the matrix forms a connected flow region unobstructed by fractures. The MINC model on the other hand applies when matrix blocks are disconnected from one another by the presence of through-going fractures. Thus the two models are complementary to one another. In particular, the MINC model is not a generalization of the DCM, but is applicable to large-scale fractures in contrast to the DCM which is applicable to rocks with a high fracture density such as characterize parts of YM. Future models for YM could employ the MINC for large-scale fractures and use the DCM to represent matrix blocks within the MINC formulation.

The added capability of the DCM in MULTIFLO will enable evaluation of DOE's current DCM modeling effort. Because many of the thermal-hydrologic aspects of DOE's Total System Performance Assessment for the Viability Assessment will be based on the DCM rather than the ECM, it is important for the CNWRA to also have an independent capability to conduct effective reviews. This is especially true because of the greater flexibility and additional parameter requirements of the DCM.

## 2.3 CURRENT STATUS OF MULTIFLO

The current status of MULTIFLO is as follows:

- Unstructured grid has been completed in METRA and GEM but is not fully tested
- Programming MINC into METRA has been initiated, but not completed
- Operator splitting is not implemented with the unstructured grid version of GEM
- The MULTIFLO User's Manual is incomplete with respect to MINC and the unstructured grid

## 2.4 DUAL CONTINUUM MODEL IMPLEMENTATION IN MULTIFLO

The DCM represents a fractured porous medium as two interacting continua: one continuum represents the fracture network and the other the rock matrix. In the case of solute transport a linear coupling term describes mass transfer between the two continua. For partially saturated systems the coupling is a nonlinear function of the saturation and is linear in the pressure difference between matrix and fracture network. The DCM is presumed valid provided the rock mass contains fractures which are connected to form a continuous flow network, typical of rock with a high density of fractures which are closely spaced. The matrix must also form a connected flow regime. For a system with widely spaced continuous fractures which isolate matrix blocks thereby disrupting their continuity, the dual continuum approach is not valid and an explicit representation of each fracture or a multiple interacting continua model MINC approach may be necessary.

Flow equations for the DCM consist of separate mass conservation equations for the matrix and fracture. As a result it necessary to solve twice the number of equations compared to a single continuum model. The implementation of the DCM into MULTIFLO is relatively easy because the complete structure for a single continuum is already in place. Furthermore, the coupling terms are linear in pressure or concentration difference between matrix and fracture.

## 2.5 RELATED CODES

The code DCM3D (Updegraff et al., 1991) applies the DCM model to unsaturated flow. However, the code applies only to isothermal conditions and uses incorrect coupling terms between fracture and matrix which depend only on matrix and not fracture properties. Codes used by LBNL which incorporate the DCM, such as TOUGH and its derivatives (Pruess, 1989), are not currently available. The NUFT code (Nitao, 1996) incorporates the same general approach to the DCM as envisaged for MULTIFLO. This code may be available in the future for comparison and benchmarking with MULTIFLO.

## 3 IMPLEMENTATION

It is proposed to begin work on the DCM immediately, postponing further work on MINC until the DCM is completed and implemented for the following reasons:

- An immediate need exists for an alternative model to replace the ECM
- The DCM requires far less programming effort compared to the MINC implementation
- MINC may not be applicable to small-scale fractures at YM which constitute the bulk of the rock mass. This type of geometry may be better described by the DCM

An estimate of time and effort involved in the planned developmental work and the order in which the work will be performed is provided in table 3-1. Although, the time for some tasks may exceed the individual estimates, the total time should represent a good estimate. This time includes debugging time which adds additional uncertainty. Work will be carried out by M. Seth under supervision by P. Lichtner.

**Table 3-1. Estimation of DCM implementation time**

Task	Description	Time (hrs)	
		METRA	GEM
I	DCM Coding	60	60
II	Testing	40	40
III	Revise User's Manual	16	16
Total		116	116

## 4 REFERENCES

- Nitao, J.J. 1996. *Reference Manual for the NUFT Flow and Transport Code, Version 1.0*. UCRL-ID-113520. Lawrence Livermore National Laboratory: Lawrence Livermore, CA.
- Pruess, K., and T.N. Narisimhan. 1985. A practical method for modeling fluid and heat flow in fractured porous media. *Society of Petroleum Engineers* 25(1): 14-27.
- Pruess, K. 1991. *TOUGH2: A General—Purpose Numerical Simulator for Multiphase Fluid and Heat Flow*. LBL-29400. Lawrence Berkeley Laboratory: Berkeley, CA.
- Updegraff, C.D., C.E. Lee, D.P. Gallego. 1991. *DCM3D: A Dual-Continuum, Three-Dimensional, Groundwater Flow Code for Unsaturated, Fractured Porous Media*. NUREG/CR-5536. Washington, DC: Nuclear Regulatory Commission.

15/25

# CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

## DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE: MULTIFLO Version 1.2

1 of 4 pages

March 2, 2000

### MULTIFLO (Scientific and Engineering Software) Version 1.2

**NOTE:** This version of the MULTIFLO Software is a follow on to the Beta version and contains changes to that Beta version released.

1. **This Design Verification Report is prepared by:** Bruce Mabrito in conjunction with the CNWRA Software Developer, Scott Painter.  
**Full Title of CNWRA scientific and engineering software:** MULTIFLO Version 1.2.  
**Demonstration work station:** A Pentium II NT computer telnetted to the VULCAN server.  
**Operating System:** SunOS 5.6

2. **Software Requirements Description and any changes thereto approved by Element Manager?**  
☒ YES                      NO                      N/A

Note: The SRD was for the MULTIFLO Version 1.2 code and was dated March 1998.

3. **Software Development Plan (SDP) and any changes have been approved by the Element Manager?**  
YES                      ☒ NO                      N/A

Note: No SDP was utilized because the original SRD predates the need for an SDP.

4. **Design and Development**  
**Module-level testing is documented in either scientific notebooks or in Software Change Reports?**  
☒ YES                      NO                      N/A

Note: Testing is documented in Scientific Notebook No. 282 which is maintained by S. Painter.

5. **Is the CNWRA scientific and engineering software developed in accordance with the conventions described in the SDP?**  
YES                      NO                      ☒ N/A

Note: This MULTIFLO code predates the need for an SDP, however the general conventions were followed.

2 of 4 pages

16/25

6. Is the CNWRA software documented internally?  
☒ YES NO N/A

Does the primary program header contain the following information:

A. Program title, Developed for (Customer), Office/Division/Date/Customer Contact/Telephone number, Software Developer, Telephone number, titles of Associated Documentation/Designator, and the Disclaimer Notice?

☒ YES NO N/A

B. Source code module header information provides Program Name, Client Name, Contract Reference, Revision number?

☒ YES NO N/A

Note: N/A

7. Software designed so that individual runs are uniquely identified by Date, Time, Name of software and version? ☒ YES NO N/A

8. The physical labeling on the software or the referenced list has Program Name/Title, Module/Name/Title, Module Revision, File Type (i.e. ASCII, OBJ, EXE), Recording Date and Operating System of the Supporting Hardware?

☒ YES NO N/A

9. Users' Manual

Is there a Users' Manual for the software?

☒ YES NO N/A

Note: There has been produced a CNWRA Users' Manual and the MULTIFLO Revision 2, Change 2 version was sent out from the CNWRA in January 2000.

Are there basic instructions for the use of the software?

☒ YES NO N/A

Note: N/A



**10. Acceptance Testing**

**Does the acceptance testing demonstrate whether or not requirements in the SRD/SDP have been fulfilled?**

**YES**

NO

N/A

Note: Requirements in the SRD were met and the desired results have been included in Scientific Notebook 282.

**Has acceptance testing been conducted for each intended computer platform and operating system?**

**YES**

NO

N/A

Note: The acceptance testing for MULTIFLO Version 1.2 was conducted by S. Painter and M. Seth on three different platforms, one of those an NT system.

**Have installation tests been performed on the target platform?**

**YES**

NO

N/A

Note: N/A

**11. Configuration Control**

**Is the Software Summary Form completed and signed?**

**YES**

NO

N/A

Note: N/A

**12. Is a software technical description prepared, documenting the essential mathematical and numerical basis?**

**YES**

NO

N/A

Note: Parts of the MULTIFLO Version 1.2 technical description is in the Users' Manual and the SRD. In addition, Chapter 1 in the book Reactive Transport in Porous Media/Reviews in Mineralogy, Vol. 34, written by P.C. Lichtner, C.I Steefel, and E.H. Oelkers, provides a mathematical and numerical basis of this code.

**13. Is the source code available (or, is the executable code available in the case of commercial codes)?**

**YES**

NO

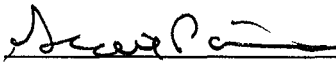
N/A

Note: The MULTIFLO Version 1.2 source code will be provided to the NRC (B. Leslie) and they may allow the DOE to use it on the Yucca Mountain Project.


18/25

14. Have all the script/make files and executable files been submitted to the Software Custodian?  
YES NO N/A

Note: All make files will be on the CD that is turned into CNWRA Quality Assurance for retention.

  
\_\_\_\_\_  
Scott Painter Date  
CNWRA MULTIFLO Software Developer

3/2/00

  
\_\_\_\_\_  
Bruce Mabrito Date  
CNWRA Software Custodian

3/2/00

Attachments/

Original to: Software Folder  
cc: CNWRA Software Developer  
Cognizant EM

19/25

c Copyright 2000 Southwest Research Institute

c Program Name: MULTIFLO

c File/Program Name: mainmli.f/MULTIFLO

c Other modules: block data metragem

c cputim.f

c seconds.f

c lnblk.f

c convert.f

c frfmt.f

c Release Date: February 2000

c Release Version: 1.2

c Client Name: USNRC

c Client Contact: Bret Leslie (301-415-6652)

c Contract Number: NRC 02-97-009

c CNWRA Contact: Scott Painter (210-522-3348)

c Center for Nuclear Waste Regulatory Analyses

c San Antonio, Texas 78238-5166

c spainter@swri.edu

cc

# c VERSION/REVISION HISTORY

c \$Id\$

c \$Log\$

c-----  
c Date Author(s) Comments/Modifications  
c-----

c April 97 Peter C. Lichtner Initial Implementation

c Mohan S. Seth

cc

# c DISCLAIMER/NOTICE

c This computer code/material was developed as an account of work  
c performed by the Center for Nuclear Waste Regulatory Analyses (CNWRA)  
c for the Division of Waste Management of the Nuclear Regulatory  
c Commission (NRC), an independent agency of the United States  
c Government. The developer(s) of the code nor any of their sponsors  
c make any warranty, expressed or implied, or assume any legal  
c liability or responsibility for the accuracy, completeness, or  
c usefulness of any information, apparatus, product or process  
c disclosed, or represent that its use would not infringe on  
c privately-owned rights.

c IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW WILL THE SPONSORS  
c OR THOSE WHO HAVE WRITTEN OR MODIFIED THIS CODE, BE LIABLE FOR  
c DAMAGES, INCLUDING ANY LOST PROFITS, LOST MONIES, OR OTHER SPECIAL,  
c INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR  
c INABILITY TO USE (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA  
c BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY THIRD PARTIES OR A  
c FAILURE OF THE PROGRAM TO OPERATE WITH OTHER PROGRAMS) THE PROGRAM,  
c EVEN IF YOU HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES,  
c OR FOR ANY CLAIM BY ANY OTHER PARTY.



21/  
25

c DISCLAIMER/NOTICE

c This computer code/material was developed as an account of work  
c performed by the Center for Nuclear Waste Regulatory Analyses (CNWRA)  
c for the Division of Waste Management of the Nuclear Regulatory  
c Commission (NRC), an independent agency of the United States  
c Government. The developer(s) of the code nor any of their sponsors  
c make any warranty, expressed or implied, or assume any legal  
c liability or responsibility for the accuracy, completeness, or  
c usefulness of any information, apparatus, product or process  
c disclosed, or represent that its use would not infringe on  
c privately-owned rights.

c IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW WILL THE SPONSORS  
c OR THOSE WHO HAVE WRITTEN OR MODIFIED THIS CODE, BE LIABLE FOR  
c DAMAGES, INCLUDING ANY LOST PROFITS, LOST MONIES, OR OTHER SPECIAL,  
c INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR  
c INABILITY TO USE (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA  
c BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY THIRD PARTIES OR A  
c FAILURE OF THE PROGRAM TO OPERATE WITH OTHER PROGRAMS) THE PROGRAM,  
c EVEN IF YOU HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES,  
c OR FOR ANY CLAIM BY ANY OTHER PARTY.

[illegible]

c PURPOSE:

c This routine is the main program for the MULTIFLO driver which  
c couples METRA and GEM.

[illegible]

23/25

```

c-----
c              definition of unit numbers
c-----
c      unit      #      description      unit      #      description
c-----
c      iunit1    7      masin read      iunit11  17      aq-brk
c      iunit2    8      masout output   iunit12  18      aq-ini
c      iunit3    9      aq-primary      iunit13  19      flow-x
c      iunit4   10      aq-secondary   iunit14  20      capznb.xyp flow
c      iunit5   11      database        iunit15  21      heat-out
c      iunit6   12      min-rate        iunit16  22      heatsrc.in
c      iunit7   13      min-vol         iunit17
c      iunit8   14      min-znbnd       iunit18
c      iunit9   15      min-sat         iunit19
c      iunit10  16      min-surf        iunit20
c                                     iunit33  33      aq-total
c                                     iunit34  34      elec
c                                     iunit35  35      sorb
c                                     iunit37  37      gas
c-----

```

*Example of comment  
lines.*

```
c-----  
c      icode = 1  metra  
c          = 2  gem  
c          = 3  gem + metra  
c          = 4  gem + metra (steady-state)  
c-----  
  
      if(icode.gt.1) then  
        call gunits (icode)  
        write(iunit2,*) 'This file was created on: ',fdate()  
      endif
```



$$25/25^-$$

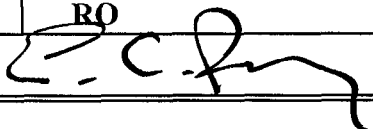
VERSION 1.2

MULTIPHASE-MULTICOMPONENT CHEMICAL TRANSPORT MODEL

Page 1

**MULTIFLO**  
**V.1.2.1**

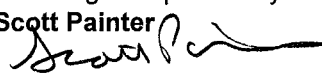
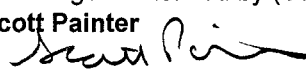
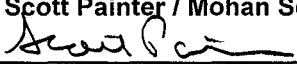
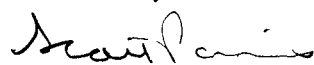
# SOFTWARE RELEASE NOTICE

1. SRN Number: <b>PA-SRN-221</b>		
2. Project Title: <b>Near-Field Environment KTI</b>		Project No. <b>20-1402-752</b>
3. SRN Title: <b>MULTIFLO Version 1.2.1</b>		
4. Originator/Requestor: <b>Bruce Mabrito</b>		Date: <b>5/10/2000</b>
5. Summary of Actions <ul style="list-style-type: none"> <li><input type="checkbox"/> Release of new software</li> <li><input checked="" type="checkbox"/> Release of modified software:             <ul style="list-style-type: none"> <li><input type="checkbox"/> Enhancements made: DCM, (Unsturctured grid.)</li> <li><input checked="" type="checkbox"/> Corrections made: Bugs corrected.</li> </ul> </li> <li><input type="checkbox"/> Change of access software</li> <li><input type="checkbox"/> Software Retirement</li> </ul>		
6. Persons Authorized Access		
Name	Read Only/Read-Write	Addition/Change/Delete
Scott Painter	RW	Addition/Change/Delete
Debra Hughson	RO	
Lauren Browning	RO	
Ron Green	RO	
Goodluck Ofoegbu	RO	
Melissa Hill	RO	
Walter Illman	RO	
Mohan Seth	RW	
English Percy	RO	
Bret Leslie	RO	
Hans Arlt	RO	Addition/Change/Delete
7. Element Manager Approval: 		Date: <b>5/19/2000</b>
8. Remarks: <b>NONE.</b>		

SOFTWARE SUMMARY FORM

01. Summary Date: <b>05/10/2000</b>	02. Summary prepared by (Name and phone) <b>Scott Painter, 522-3348</b>	03. Summary Action:  <b>New</b>	
04. Software Date: <b>04/17/2000</b>	05. Short Title: <b>MULTIFLO Version 1.2.1</b>		
06. Software Title:  <b>MULTIFLO Version 1.2.1</b>		07. Internal Software ID:  <b>NONE</b>	
08. Software Type:  <input type="checkbox"/> Automated Data System  <input checked="" type="checkbox"/> Computer Program  <input type="checkbox"/> Subroutine/Module	09. Processing Mode:  <input type="checkbox"/> Interactive  <input type="checkbox"/> Batch  <input checked="" type="checkbox"/> Combination	10. APPLICATION AREA a. General: <input checked="" type="checkbox"/> Scientific/Engineering <input type="checkbox"/> Auxiliary Analyses <input type="checkbox"/> Total System PA <input type="checkbox"/> Subsystem PA <input type="checkbox"/> Other  b. Specific: <b>Groundwater multiphase flow and reactive transport model</b>	
11. Submitting Organization and Address: <b>CNWRA 6220 Culebra Road San Antonio, TX 78228</b>		12. Technical Contact(s) and Phone:  <b>Scott Painter, (210) 522-3348 Mohan Seth, (972) 699-3610</b>	
13. Narrative: <b>The code is used to model multiphase groundwater flow and reactive transport.</b>			
14. Computer Platform  <b>SUN</b>	15. Computer Operating System:  <b>UNIX</b>	16. Programming Language(s):  <b>Fortran 77</b>	17. Number of Source Program Statements:  <b>~80,000</b>
18. Computer Memory Requirements: <b>Problem Dependent</b>	19. Tape Drives:  <b>N/A</b>	20. Disk/Drum Units:  <b>N/A</b>	21. Graphics:  <b>ASCII plot data files</b>
22. Other Operational Requirements <b>Thermodynamic database required.</b>			
23. Software Availability:  <input checked="" type="checkbox"/> Available <input type="checkbox"/> Limited <input type="checkbox"/> In-House ONLY		24. Documentation Availability:  <input checked="" type="checkbox"/> Available <input type="checkbox"/> Inadequate <input type="checkbox"/> In-House ONLY <b>DRAFT</b>	
Software Developer: <i>Scott Painter</i> Date: <b>5-10-00</b>			

# SOFTWARE CHANGE REPORT (SCR)

1. SCR No. (Software Developer Assigns): <b>PA-SCR-314</b>	2. Software Title and Version: <b>Multiflo V.1.2.1</b>	3. Project No: <b>20.01402.562</b>
<p>4. Affected Software Module(s), Description of Problem(s):</p> <p><b>METRA module was failing at end of run when Liquid only keyword was activated. Some files had incorrect dates and version numbers.</b></p>		
<p>5. Change Requested by:</p> <p><b>Scott Painter</b></p>  <p>Date: <b>March 17, 2000</b></p>	<p>6. Change Authorized by (Software Developer):</p> <p><b>Scott Painter</b></p>  <p>Date: <b>March 17, 2000</b></p>	
<p>7. Description of Change(s) or Problem Resolution (If changes not implemented, please justify):</p> <p><b>Updated version numbers and dates.</b></p> <p><b>Re-coded air and mass balance calculation in the emp subroutine located in file metra/emp.f.</b></p> <p><b>Made minor improvements to internal documentation and changed dimensioning of aa array in GEM to be consistent with METRA.</b></p> <p><b>Corrected call to RADCORD that had wrong number of arguments. This involved changes to mainmti.f, gem/maingem.f, gem/pprcgem.f, gem/readgem.f, metra/mainmetra.f, metra/pproc.f.</b></p>		
<p>8. Implemented by:</p> <p><b>Scott Painter / Mohan Seth</b></p> 	<p>Date:</p> <p><b>May 16, 2000</b></p> <p><i>S</i></p>	
<p>9. Description of Acceptance Tests:</p> <p><b>Ran saturated drawdown problem (Theis's problem). Code executed correctly and produced results identical to benchmark problem. Also performed limited regression testing. Verified that correct version and date was printed.</b></p>		
<p>10. Tested by: <b>Scott Painter</b></p> 	<p>Date: <b>May 16, 2000</b></p>	

# CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

5/32

## DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE: MULTIFLO Version 1.2.1

Date: May 19, 2000

1. This Design Verification Report is prepared by Scott Painter and Bruce Mabrito in accordance with TOP-018, Development and Control of Scientific and Engineering Software, section 5.8.

Full Title of CNWRA scientific and engineering software: MULTIFLO Version: 1.2.1

Software Category (See TOP-018, Table 1): Developed

Demonstration workstation: PC/SUN/VULCAN Server

Operating System: UNIX/SOLARIS

2. Software Requirements Description and any changes thereto follow QAP-002 requirements?

YES

NO

N/A

Notes: The March 1998 Software Requirements Description for MULTIFLO V. 1.2 was used. It is located in the MULTIFLO V. 1.2 Software folder in the QA Records Room.

3. The Element Manager has approved the Software Development Plan (SDP) and any changes?

YES

NO

N/A

Notes: MULTIFLO V. 1.2 predated the Software Development Plan requirements.

4. Design and Development

Module-level testing is documented either in scientific notebooks or in Software Change Reports?

YES

NO

N/A

Notes: PA-SCR-314 is the Software Change Report that was used for this change.

5. Is the CNWRA scientific and engineering software developed in accordance with the conventions described in the SDP?

YES

NO

N/A

Notes: S. Painter utilized the original SRD and made changes identified on the Software Change Report.

# CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

6/32

## 6. Is the CNWRA software documented internally?

☒ YES

NO

N/A

Notes: Examples of internal documentation are attached.

### Does the primary program header contain the following information?

A. Program title, Developed for (Customer), Office/Division/Date/Customer Contact/Telephone number, Software Developer, Telephone number, titles of Associated Documentation/Designator, and the Disclaimer Notice?

☒ YES

NO

N/A

B. Source code module header information provides Program Name, Client Name, Contract Reference, Revision number?

☒ YES

NO

N/A

## 7. Software designed so that individual runs are uniquely identified by Date, Time, Name of software and version?

☒ YES

NO

N/A

Notes: Examples are attached.

## 8. The physical labeling on the software or the referenced list has Program Name/Title, Module/Name/Title, Module Revision, File Type (i.e. ASCII, OBJ, EXE), Recording Date and Operating System of the Supporting Hardware?

☒ YES

NO

N/A

Notes: None

## 9. Users' Manual

### Is there a Users' Manual for the software?

☒ YES

NO

N/A

If no, explain: The MULTIFLO V. 1.2 User's Manual is in the QA Records Room.

### Are there basic instructions for the use of the software?

☒ YES

NO

N/A

Notes: Contained in MULTIFLO V. 1.2 User's Manual.

# CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

47/32

## 10. Acceptance Testing

Does the acceptance testing demonstrate whether or not requirements in the SRD have been fulfilled?

YES

NO

N/A

Notes: The Software Change Report describes the changes and the acceptance testing.

Has acceptance testing been conducted for each intended computer platform and operating system?

YES

NO

N/A

Notes: Windows NT and SOLARIS were platforms on which they were tested, both of which the NRC could use at their offices.

Have installation tests been performed on the target platform?

YES

NO

N/A

Notes: None.

## 11. Configuration Control

Is the Software Summary Form completed and signed?

YES

NO

N/A

If no, explain: N/A

12. Is a software technical description prepared, documenting the essential mathematical and numerical basis?

YES

NO

N/A

If no, explain: See MULTIFLO V. 1.2 User's Manual.



# CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

8/32

13. Is the source code available (or, is the executable code available in the case of (acquired/commercial codes)?

YES

NO

N/A

Notes: S. Painter and M. Seth have read/write authority, and NRC staff have been included, as noted in the Software Release Notice.

14. Have all the script/make files and executable files been submitted to the Software Custodian?

YES

NO

N/A

Notes: See the CD in the MULTIFLO Version 1.2.1 folder.

  
\_\_\_\_\_  
CNWRA Software Developer      5/19/2000  
Scott Painter                      Date

  
\_\_\_\_\_  
CNWRA Software Custodian      5/19/2000  
Bruce Mabrito                      Date

Attachments/

Original to: Software Folder  
cc: CNWRA Software Developer  
Cognizant EM

```
head -20 dcm1.out
```

VERSION 1.2.1

MULTIPHASE-MULTICOMPONENT CHEMICAL TRANSPORT MODEL

dcm demonstration with YM parameters

4 - ©

3 Sess-1

129.162.200.176

24/9

Date: 05/20/00 Time: 11:35:10

← This is reflecting "one day off"  
AS IT WAS PRINTED 5/19/2000.  
Sen QA



C EVEN IF YOU HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES,  
C OR FOR ANY CLAIM BY ANY OTHER PARTY.

[illegible]

C PURPOSE:

```
c  This routine is the main program for the MULTIFLO driver which
c  couples METRA and GEM.
```

[illegible]

### C INTERFACING ARGUMENTS:

c	Variable name	Type	Description
c	=====	=====	=====
c	none		

```

c-----
c  Externals
c  =====

```

c none

[illegible]

## C INTERFACING ROUTINES

```
c      Calling routines
c      =====
```

c none

c	Called routines	Function
c	=====	=====

```
c      gunits.f      -Command line interface for gem.
c      metra.f      -Driver for METRA (Similar to stand-alone
c                   main program.
c      gem.f        -Driver for GEM (Similar to stand-alone
c                   main program.
c      cputim.f     -Routine to measure cpu time.
```

[illegible]

## C INCLUDE FILES

```

c      Name      Description
c      =====

```

```
c      include files common to metra and gem
c      met/impl.h          -Declares real variables to real*8 and sets
c                          frequently used constants in common.
c      met/metragem.h      -Variables which are common to both metra and
c                          gem codes.
c
c      met/paramtrs.h      -Sets dimension limits for all variables.
c
c      gem include files
c      gem/addgem.h        -Sets pointers for dynamic memory.
c      gem/scalgem.h       -Scalars in common.
c      gem/comqgem.h       -General common block.
```

```
c    gem/iounits.h    -I/O unit numbers.
```

[illegible]

## C SYSTEM LIBRARY ROUTINES

C	Name	Description
C	=====	=====

```
c      etime, fdate()
```

[illegible]

```
C  OUTPUT UNIT(S)
```

c	Unit Name(Number)	Description	file name
---	-------------------	-------------	-----------

```
c      iunit2 (8)      normal run output      masout
```

[illegible]

## C REFERENCES

c none

[illegible]

### C PROGRAM FLOW

```

c                                     Begin
c                                     |
c -----
c                                     if (icode .gt. 1) call gunits
c -----
c                                     |
c -----
c                                     if (icode .ne. 2) call metra
c                                     <if icode = 1, code does not return from metra>
c -----
c                                     |
c -----
c                                     call gem
c                                     <if icode = 4, code does not return from gem>
c -----
c                                     |
c -----
c                                     if (icode .eq. 3) call allot
c -----
c                                     |
c -----
c                                     <begin time stepping loop for icode = 3>
c                                     120 continue
c                                     tmetra = tmetra + dtnew
c -----
c                                     |
c -----
c                                     call metra
c -----
c                                     |
c -----
c                                     call gem
c -----

```

```

c          |
c          -----
c          if (nprint .gt. nprin) goto 190
c          -----
c          |
c          -----
c          goto 120
c          -----
c          |
c          -----
c          190 continue
c          -----
c          |
c          -----
c          STOP
c          -----

```

# program multiflo

```

c*****
c
c  multiflo is a multi-component-multi-phase chemical transport
c  model for hydrothermal fluids reacting with minerals. this version
c  of multiflo was completed August 1995.
c
c  authors ... peter c. lichtner and mohan seth
c              cnwra
c              southwest research institute
c              san antonio, texas
c
c  description ... multiflo simulates solute transport for steady
c                   one-dimensional mass transport by advection,
c                   dispersion and diffusion in a saturated porous
c                   medium.
c                   chemical reactions incorporated in the code
c                   include aqueous complexation, redox reactions,
c                   precipitation/dissolution of minerals and ion
c                   exchange. provision is included for both
c                   reversible and irreversible reactions of minerals.
c
c  main program:
c  mainmlti.f
c
c  subroutines:
c
c  GEM
c  allotgem.f      difoft.f      gridld.f      mastrnos.f      solver.f
c  blkdtgem.f      eqjac.f        gunits.f      maxchg.f        speciate.f
c  bndcond.f       eqlib.f        hybrid.f      mltpsiex.f      startup.f
c  cehyliq.f       egres.f        initgem.f     modbnd.f        stdyst.f
c  cehytwph.f      explicit.f     initrate.f    mprove.f        stepgem.f
c  cetvdlq.f       fit.f          interpf.f     opspltex.f      testgem.f
c  cetvdtwp.f      fkinet.f        ionexc.f      opspltgl.f      textab.f
c  cexact.f        flogk.f        kinrxn.f      opspltim.f      transp.f
c  cgasos.f        fun.f          output1.f     unitconv.f
c  cihytwph.f      gameq.f        kinrxnex.f    output2.f      updtgem.f
c  cliqos.f        gamext.d.f    lubksb.f      path.f          watsolv.f
c  coefrxn.f       ghostpsi.f    ludcmp.f      pecletrn.f      zonek.f
c  coshyliq.f      graphld.f    maingem.f     psat.f

```

```

c dataall.f      graph2d.f      massbal.f      readat.f
c database.f     graph3d.f      mastrnex.f     solprd.f
c density.f      grid.f      mastrnim.f     solprodt.f

c      include files
c addgem.h      fields.h      iounits.h      ofiles.h      surfkin.h
c comgem.h      frfmt.h      kinetic.h      paramtrs.h    tempfld.h
c cxkin.h       gas.h      metragem.h     scalgem.h     velsat.h
c debye.h       impl.h      minrl.h       scratch.h     watsolv.h

c      THERMODYNAMIC DATABASE
c      ms25.r16
c      mstemp.r16

c      METRA

c      subroutines
c accm.f        debug.f      mainmetra.f    pvth2o.f      trans.f
c accmvp.f      dtstep.f     misc.f        pvtvp.f      update.f
c allot.f       ecmtbl.f    openfls.f     recdat.f     updtpsk.f
c bcond.f       emip.f      outmetra.f    rstart.f     updtvpk.f
c blkdtmet.f    equil.f      pckr.f        setbc.f      watsolv.f
c coefs.f       griddat.f    plots.f       slvliq.f
c coefsvp.f     init.f      prints.f      solve.f
c cond.f        inpmetra.f  pvt.f         source.f
c d4gaus.f      iter.f      pvtfunc.f     thomas.f

c      include files
c add.h         frfmt.h      metragem.h    pckr.h        pvttbl.h      units.h
c com.h         impl.h      paramtrs.h    pvtfunc.h     scalars.h     watsolv.h

C=====

      include 'metra/impl.h'
      include 'metra/paramtrs.h'
      include 'metra/metrage.m.h'
      include 'metra/units.h'

      include 'gem/addgem.h'
      include 'gem/scalgem.h'
      include 'gem/comgem.h'
      include 'gem/iounits.h'

C=====
c      Include files:impl.h, paramtrs.h, metragem.h, watsolv.h and frfmt.h
c      are used both by METRA and GEM, and must be identical.
c      Subroutines:  frfmt.f and watsolv.f are used by both METRA and GEM.
C=====

      real etime,tyme(2)

      character*24 fdate

      save maxaa,icodsav,time1,time2

      common/comaa/aa(5000000)
      common/cpus/ cpusub(30),tim1,tim2,timmetra,timgem

      data fmwh2o,cpusub/18.016d0,30*0.d0/

      maxaa =5000000

```

definition of unit numbers						
	unit	#	description	unit	#	description
	iunit1	7	masin read	iunit11	17	aq-brk
	iunit2	8	masout output	iunit12	18	aq-ini
	iunit3	9	aq-primary	iunit13	19	flow-x
	iunit4	10	aq-secondary	iunit14	20	capznb.xyp flow
	iunit5	11	database	iunit15	21	heat-out
	iunit6	12	min-rate	iunit16	22	heatsrc.in
	iunit7	13	min-vol	iunit17		
	iunit8	14	min-znbnd	iunit18		
	iunit9	15	min-sat	iunit19		
	iunit10	16	min-surf	iunit20		
				iunit33	33	aq-total
				iunit34	34	elec
				iunit35	35	sorb
				iunit37	37	gas

```
call cputim (-1)

cpu = etime(tyme)
ibgn = 1

write(*,'(''--> read icode: (1-metra, 2-gem, 3-coupled, '',  
.'''4-coupled: steady-state) ''',$)'')  
read(*,*) icode

icodsav = icode

c-----  
c      icode = 1   metra  
c           = 2   gem  
c           = 3   gem + metra  
c           = 4   gem + metra (steady-state)  
c-----  
  
if(icode.gt.1) then  
    call gunits (icode)  
    write(iunit2,*) 'This file was created on: ',fdate()  
endif  
C*****  
c      include 'title.ins'          ! print title etc.  
C*****  
if(icode .ne. 1) then  
write(*,'(/,8x,66(''_-''))'  
write(*,*)  
write(*,'(10x,  
. '_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_ _/_/_ _''  
)')  
write(*,'(10x,  
. '_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_ _/_/_ _''  
)')  
write(*,'(10x,  
. '_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_ _/_/_ _''  
)')  
write(*,'(10x,  
. '_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_ _/_/_ _''
```



```

.)')
write(*,'(10x,
.' _/ _/ _/_/_/ _/_/_/ _/ _/ _/_/_/ _/_/ ''
.)')
write(*,*) ' '
write(*,'(20x,' ' Developed for the U.S. NRC''')')
write(*,*) ' '
write(*,'(20x,' ' Version 1.2.1 ''')')
write(*,*) ' '
write(*,'(20x,' ' May, 2000''')')
write(*,*) ' '
write(*,'(16x,' 'MULTIPHASE-MULTICOMPONENT CHEMICAL TRANSPORT''',
.' ' MODEL''')')
write(*,*)
write(*,'(18x,
.' 'Copyright (c) 2000 Southwest Research Institute''')')
write(*,'(18x,' ' All Rights Reserved''')')
write(*,'(8x,66(''_')/))')
C*****
write(iunit2,*) ' '
write(iunit2,'(10x,
.' _/ _/ _/ _/ _/_/_/_/_/ _/ _/_/_/_/ _/ _/_/ ''
.)')
write(iunit2,'(10x,
.' _/_/_/ _/_/_/ _/ _/ _/ _/ _/ _/ _/ ''
.)')
write(iunit2,'(10x,
.' _/ _/ _/ _/ _/ _/ _/ _/_/_/ _/ _/ _/ ''
.)')
write(iunit2,'(10x,
.' _/ _/ _/ _/ _/ _/ _/ _/ _/ _/ _/ _/ ''
.)')
write(iunit2,'(10x,
.' _/ _/ _/_/_/ _/_/_/ _/ _/ _/ _/_/_/ _/_/ ''
.)')
write(iunit2,*) ' '
write(iunit2,'(20x,' ' Developed for the U.S. NRC''')')
write(iunit2,*) ' '
write(iunit2,'(20x,' ' VERSION 1.2.1''')')
write(iunit2,*) ' '
write(iunit2,'(20x,' ' May, 2000''')')
write(iunit2,*) ' '
write(iunit2,'(16x,' 'MULTIPHASE-MULTICOMPONENT CHEMICAL ''',
.' 'TRANSPORT MODEL''')')
write(iunit2,*)
write(iunit2,*)
.' ' Copyright (c) 2000 Southwest Research Institute'
write(iunit2,*) ' All Rights Reserved'
write(iunit2,*)
end if

C*****
C*****
c initialize running processes
C*****
igeom = 0 !initialize

if (icode.ne.2) then

write(*,*) '--> Initializatzize METRA'

```

```

    iperm = 0

    call seconds(time1)

    call metra (dtmetra,dtnew,tmetra,0.d0,aa,maxaa)

    call seconds(time2)
    cpusub(10) = cpusub(10) + time2-time1

    if(icode.eq.1) go to 190
c    change the density units to gms/cm3 from mole/m3

    cvfac1 = fmwh2o*1.d-3
    cvfac2 = one/cvfac1

    do m = 1,nb
        rho2(m) = dwk(m)*cvfac1 ! rho2 is equivalenced with dwk
    end do
endif

write(*,*) '--> Initializatzize GEM'

dtmetra = zero
call seconds(time1)

call gem (aa,maxaa)

call seconds(time2)
cpusub(25) = cpusub(25) + time2-time1

if(icode.eq.2) go to 190

c                                reset the addresses for metra
if(icode.eq.3) then
c    ibgn = ir    ! check this later..it should work-save much space
    ibgn = iaa
    itime = 1
    call allot (aa,3)
endif

c ***** initialization section ends *****

    dtmetra = dt(1)
    dtnew   = dtmetra
    tmetra  = zero

c*****
c                                take new time step
c*****

120 continue

    tmetra = tmetra + dtnew

c-----adjust time step to print or plot time

    if (time(nprint) .gt. tplot(kplot)) then
        lplt = kplot
        if(tmetra.gt.tplot(kplot)) then
            tmetra = tmetra-dtnew

```

```

        dtnew = tplot(kplot)-tmetra
        tmetra = tplot(kplot)
    endif
else
    if(tmetra.eq.time(nprint)) then
        tmetra = time(nprint)
    else if(tmetra.gt.time(nprint)) then
        tmetra = tmetra-dtnew
        dtnew = time(nprint)-tmetra
        tmetra = time(nprint)
    endif
endif
endif

c-----
c  convert liquid density from gms/cm3 (kgms/ltr) to Moles/m3

    do m = 1,nb
        dwk(m) = rho2(m)*cvfac2 ! rho2 is equivalence with dwk
    end do

    call seconds(time1)
    call metra (dtmetra,dtnew,tmetra,tplot(kplot),aa,maxaa)
    call seconds(time2)
    cpusub(10) = cpusub(10) + time2-time1

c  convert liquid density from Moles/m3 to gms/cm3 (kgms/ltr)

    do m = 1,nb
        rho2(m) = dwk(m)*cvfac1 ! rho2 is equivalence with dwk
    end do

c          reset the time before incrementing.

c-----
    call seconds(time1)
    call gem (aa,maxaa)
    call seconds(time2)
    cpusub(25) = cpusub(25) + time2-time1
c-----

c          stop if last time step reached
    if(tmetra .ge. time(nprint)) then
        nprint = nprint + 1
        if (nprint .gt. nprin) goto 190
    endif

c          reset plotting counter
    if(tmetra.ge.tplot(kplot)) then
        kplot=kplot+1
        if (kplot .gt. kpltmax) kplot = kpltmax
    endif
    go to 120
c-----

190  continue

    if(icodsav.eq.1) then
        call cputim (28)
    else if(icodsav.eq.2) then
        call cputim (29)
    endif

```

19/32

```

else
    call cputim (30)
endif

cpu      = etime(tyme)
cpu      = tyme(1)
cpumin   = cpu/60.d0
cpuhr    = cpumin/60.d0

write(iunit0,195) mcyc,itertot,cpu,cpumin,cpuhr
write(iunit2,195) mcyc,itertot,cpu,cpumin,cpuhr
write(*,195) mcyc,itertot,cpu,cpumin,cpuhr

195 format(1x,69('-'),/1x,'normal stop - run completed in ',i7,
.' steps',3x,'total iters =',i7/1x,'cpu time = ',
. f11.4,' secs',f11.4,' min',f11.4,' hours',/1x,69('-'))

stop

200 continue

c*****
c          write restart data on tape5
c*****

c      write(iunit2,*) ' restart data written on tape iunit5 '
c      call cputim (20)

stop

333 write(iunit2,*) 'error in opening files: iunit1 or iunit2'
write(*,*) 'error in opening files: iunit1 or iunit2'
stop

end

c=====
c*file radcord

subroutine radcord (dx,rb,nx,igrid,ifer)

include 'metra/impl.h'
include 'metra/units.h'

dimension dx(0:*),rb(*)

save index
data index/0/

if(index.eq.1) return

ifer = ifbug

nxpl = nx+1

rw = rb(1)
re = rb(nxpl)
if(rw.le.zero) rw = 1.d-20

do j = 1,nx
    if(dx(j).le.zero) goto 125

```

```

end do
irc = nx
goto 160

125  irc = j-1
     nrc = irc
     rci = dx(irc)
     ul = 0.95d0*re/rci
     ul = ul**(one/(nx-nrc+1))
     nrcp = irc+1
     do itr=1,10
        u2 = ul
        u3 = ul**(nx-irc)*(ul-one)*rci-re*dlog(ul)
        u4 = rci*ul**(nx-irc-1)*((ul-one)*(nx-nrc)+ul)-re/ul
        ul = ul-u3/u4
        if (abs(ul-u2).le.1.d-5) go to 140
     end do

140  do i=nrcp,nx
        dx(i) = ul*dx(i-1)
     end do
     if(ul.le.one) then
        print *, 'in init1.f, itr r(i)/r(i-1) = ',itr,ul
        write(ifer,142) itr,ul
142  format(/1x,'** Error: Bad calculated radii, r(i)/r(i-1) = ',
           i3,e12.4)
        stop
     endif
160  continue

     if(igrid.eq.2) then ! as in ctough
     do i = 2,nx
        rb(i) = half*(dx(i)+dx(i-1))
     end do
     else
     do i=2,nx
        rb(i) = (dx(i)-dx(i-1))/dlog(dx(i)/dx(i-1))
     end do
     endif

     index = 1

     return
     end
c=====
c*file dcmfrac

subroutine dcmfrac (sigmaf,areamf,aperture,key,ifout,itp,ioutpt,
                  ierr)

include 'metra/impl.h'
include 'metra/paramtrs.h'
include 'metra/metragem.h'
include 'metra/com.h'
include 'metra/frfmt.h'

include 'gem/scalgem.h'

character key*4
dimension xlm(nbm),sigmaf(*),areamf(*),aperture(*)
equivalence (xlm,rtot)

```

```

save icall
data icall/0/

c          read fracture-matrix parameters for dcm

write (ifout,117)

565 format(6i3,8f12.0)
566 format(3i5,8f12.0)
105 format(5x,6i4,9(1p12.4))
106 format(5x,3i6,6x,9(1p12.4))
56 format(5x,81(' ')/)
117 format(//1x,'*DCMPara',10x,'Fracture Characteristic Parameters',
*/19x,34('=')//,5x,81(' ')/
*5x,' i1 i2 j1 j2 k1 k2      sigmaf      areamod      xlmm      ',
*'      ylmm      zlmm      areamf      apert      porm      swm
',/
*5x,81(' '))

do mn = 1,99999
10  if(igeom.lt.0) then
      call frfmt (ithr,ifive,ieight,12,izro,izro,itp)
      read (image,566) i1,i2,j1,usigmaf,areamod,xlmm,ylmm,
*          zlmm,uporm,uswm
      else
      call frfmt (isix,ithr,ieight,12,izro,izro,itp)
      read (image,565) i1,i2,j1,j2,k1,k2,usigmaf,areamod,xlmm,ylmm,
*          zlmm,uporm,uswm
      endif
      if(i1.le.0) then
          if(ioutpt.ge.1) write (ifout,56)
          icall = 1
          return
      endif

      porm = uporm
      swm = uswm

      if(icall.gt.0.and.abs(igeom).le.1) go to 10

      if(igeom.ge.0)
      call range (i1,i2,j1,j2,k1,k2,nx,ny,nz,key,ierr)

      if(xlmm+ylmm+zlmm.le.zero) then
          write (*,108)
          write (ifout,108)
108  format(1x,'*** Error: zero matrix block size read ')
          stop
      endif

c      fractures in 3-dimens

      if(xlmm.gt.zero.and.ylmm.gt.zero.and.zlmm.gt.zero) then

          bb = xlmm+ylmm+zlmm
          cc = xlmm*ylmm+ylmm*zlmm+zlmm*xlmm
          dd = -usigmaf*xlmm*ylmm*zlmm/(1.d0-usigmaf)

          third = 1.d0/3.d0

```

```

bb3 = bb*third
u1 = cc-bb*bb3
u2 = -(dd-bb3*cc+2.d0*bb3**3)
u3 = sqrt(u2*u2+4.d0*(u1*third)**3)
apert = (0.5d0*(u2+u3))**third+(.5d0*(u2-u3))**third-bb3

u1 = one/xlmm+one/ylmm+one/zlmm
avglm = three/u1

c   fractures in 1-dimens

else if(xlmm+ylmm.le.zero) then
    apert = zlmm*usigmaf/(one-usigmaf)
    u1 = one/zlmm
    avglm = zlmm
else if(ylmm+zlmm.le.zero) then
    apert = xlmm*usigmaf/(one-usigmaf)
    u1 = one/xlmm
    avglm = xlmm
else if(xlmm+zlmm.le.zero) then
    apert = ylmm*usigmaf/(one-usigmaf)
    u1 = one/ylmm
    avglm = ylmm

c   fractures in 2-dimens

else if(xlmm.le.zero) then
    bb = ylmm+zlmm
    cc = -usigmaf*ylmm*zlmm/(1.d0-usigmaf)
    apert = 0.5d0*(sqrt(bb*bb-4.d0*cc)-bb)

    u1 = one/ylmm+one/zlmm
    avglm = two/u1
else if(ylmm.le.zero) then
    bb = xlmm+zlmm
    cc = -usigmaf*xlmm*zlmm/(1.d0-usigmaf)
    apert = 0.5d0*(sqrt(bb*bb-4.d0*cc)-bb)

    u1 = one/xlmm+one/zlmm
    avglm = two/u1

else if(zlmm.le.zero) then
    bb = xlmm+ylmm
    cc = -usigmaf*xlmm*ylmm/(1.d0-usigmaf)
    apert = 0.5d0*(sqrt(bb*bb-4.d0*cc)-bb)

    u1 = one/xlmm+one/ylmm
    avglm = two/u1
endif

if(areamod.gt.zero) then
    uarea = two*(one-usigmaf)*u1*areamod
else
    uarea = -areamod
endif

if(igeom.ge.0) then
    if(ioutpt.ge.1)
        write(ifout,105) i1,i2,j1,j2,k1,k2,usigmaf,areamod,
*                          xlmm,ylmm,zlmm,uarea,apert,porm,swm
*
        do k = k1,k2

```

```

do j = j1,j2
  do i = i1,i2
    m = i+j1(j)+k1(k)
    inddec(m) = 1
    xlm(m) = avglm
    sigmaf(m) = usigmaf
    areamf(m) = uarea
    aperture(m) = apert
    if(icode.ne.2) then
      if(porm.gt.zero) then
        fackrmf(m) = porm
      else
        fackrmf(m) = one
      endif
    endif
  enddo
enddo
enddo
else
  if(ioutpt.ge.1)
    * write(ifout,106) i1,i2,j1,usigmaf,areamod,xlmm,ylmm,
    *               zlmm,uarea,apert,porm,swm
    do m = i1,i2,j1
      xlm(m) = avglm
      sigmaf(m) = usigmaf
      areamf(m) = uarea
      aperture(m) = apert
      if(icode.ne.2) then
        if(porm.gt.zero) then
          fackrmf(m) = porm
        else
          fackrmf(m) = one
        endif
      endif
    enddo
  endif
end do

return
end

c=====
c*file frfmt
c
c      subroutine frfmt (n1,ifld1,n2,ifld2,n3,ifld3,itpp)
c=====
c
c This routine is used for free-format reading. It sets a free
c format input-line into a prespecified format which is then
c read by 'internal read' statement. It is called by input.f
c and recdat.f routines.
c
c=====
c   Date           Author(s)           Comments/modifications
c
c February 95      Mohan S. Seth        Taken from TS&E's code by MSS
c
c=====
include 'metra/impl.h'
include 'metra/frfmt.h'

```



```

include 'gem/iounits.h'

dimension images(40)
character kard1*164,kard*164,word*4

C=====

10    read (itpp,200) (kard(i:i),i=1,ncol)

        icrd = icrd+1
        if (list.eq.0) go to 30
        write (iunit2,20) icrd,(kard(i:i),i=1,ncol)

20    format (1x,i8,2h $,80a1,1h$)
30    continue

        if (kard(1:1).eq.icmt) go to 10

        do i = 1,4
            word(i:i) = kard(i:i)
        end do

        call convert(word,4)

        if(word.eq.SKIP) then
            il = -1
40    read (itpp,100) word
            il =il+1
            call convert(word,4)
            if(word.ne.NOSKIP) go to 40
            write (*,50) il
            go to 10
        else if(word.eq.NOSKIP) then
            go to 10
        endif
50    format(/1x,30(1h=)/1x,i6,' Data Lines Skipped'/1x,30(1h=)/)
100   format(a4)

        if ((kard(1:1).eq.'/' .or. kard(1:1).eq.'*' .or. kard(1:1).eq.'.')
        .and. kard(2:2).eq.' ') kard(1:1) = iblank

60    nvar = n1+n2+n3
        do i=1,n1
            images(i) = ifld1
        end do

        nfld = ifld1*n1
        if (n2.eq.0) go to 110
        i1 = n1+1
        i2 = n1+n2
        do i=i1,i2
            images(i) = ifld2
        end do

        nfld = nfld+ifld2*n2
        if (n3.eq.0) go to 110
        i1 = i2+1
        i2 = i2+n3
        do i=i1,i2
            images(i) = ifld3
        end do

```

```

nfld = nfld+ifld3*n3
110 continue

do i=1,nfld
    kard1(i:i) = iblank
end do

kard(ncol+1:ncol+1) = iblank

i2 = 1
i1 = 0
i1 = 1

do 120 m=1,nvar
    nsize = 0
    do i=i1,ncol

        if(kard(i:i).eq.icmt.or.kard(i:i).eq.'!') go to 150
        if (kard(i:i).ne.iblank) then
            nsize = nsize+1
            if(kard(i+1:i+1).eq.iblank) go to 130
        end if
    end do
    go to 150

130    i3 = images(m)
    if(nsize.gt.i3) then
        write (*,132) nsize,i3,(kard(mm:mm),mm=1,ncol)
        write (iunit2,132) nsize,i3,(kard(mm:mm),mm=1,ncol)
        stop 'reading error in frfmt.f'
    end if
    i1 = i+1
    l = i1-i2
    if (l.lt.i3) go to 140
    i2 = i1-i3
    do k=i2,i
        l1 = l1+1
        kard1(l1:l1) = kard(k:k)
    end do
    go to 120

140    l1 = l1+i3-1
    do k=i2,i
        l1 = l1+1
        kard1(l1:l1) = kard(k:k)
    end do
120    i2 = i1
150 continue

c    print *, 'nfld ',nfld
c    write (*,*) (kard1(i:i),i=1,nfld)

    write (image,200) (kard1(i:i),i=1,nfld)
200    format (163a1)

132    format(1x,'field used for the variable > internally specified',/
    *1x,'field used = ',i3,' specified = ',i3,' on the following-',/
    *' reduce field'/(1x,121a1))

c1790    format(//1x,5hcnwra,36x,37(1h=),44x,4hpage,i4/42x,1h=,35x,1h=,/

```

```

c      +37h= i n p u t   d a t a   i m a g e s   =,/42x,1h=,35x,1h=,/42x
c      +,37(1h=)//5x,4hc no,3h 1,i9,7i10,/5x,4(1h=),2h +,16(4h====,1h+),
c      +10(1h-),24herrors/warnings detected,9(1h-))
c
c      return
c      end

```

```

c*deck  convert

```

```

c
c      subroutine convert (key,nchar)
c
c      =====
c      This routine is really cute! It converts all small case letters
c      to capitals in a character string 'key' of length nchar. It
c      is currently set to a max of 10 which can be changed to larger
c      value if desired.
c      =====

```

```

c      save ich1,ich2
c      character*26 ich1,ich2
c      character*10 key
c      character*1 char
c
c      data ich1,ich2/'abcdefghijklmnopqrstuvwxyz',
c      *              'ABCDEFGHIJKLMNOPQRSTUVWXYZ'/
c
c      do 20 i = 1,nchar
c        char = key(i:i)
c        do m = 1,26
c          if(char.eq.ich1(m:m)) go to 10
c        end do
c        go to 20
c      10    key(i:i)=ich2(m:m)
c      20    continue
c      print *, 'key   = ', (key(i:i),i=1,nchar)
c      return
c      end

```

```

c*deck lnblnk

```

```

c
c      function lnblnk(b)
c      character*(*) b
c
c      lnblnk=len(b)
c      do j=len(b),1,-1
c        if(b(j:j).eq.' ') lnblnk=lnblnk-1
c      end do
c
c      return
c      end

```

```

c*deck cputim

```

```

c
c      subroutine cputim (ipgm)
c
c      include 'metra/impl.h'
c      include 'metra/units.h'
c      include 'metra/paramtrs.h'
c      include 'metra/metragem.h'
c      include 'metra/scalars.h'
c      include 'gem/iounits.h'

```

```

c      include 'gem/scalgem.h'

      common/cpus/ cpusub(30),tim1,tim2,timmetra,tingem
      common/forcpu/ mcyc, itertot, ihalcum

      dimension percentg(30),ncalls(30)
      save ncalls

c=====
c      This subroutine captures computing time for different
c      components of the total code. Subroutine named 'seconds'
c      (appended below this routine) MUST BE MODIFIED for a
c      particular computer and the operating system.
c      This subroutine captures computing time for different
c      components of the total code. Subroutine named 'seconds'
c      (appended below this routine) MUST BE MODIFIED for a
c      particular computer and the operating system.
c
c      Usage: to measure cpu time of subroutine/process
c              call cputim (0)
c              call subroutine/process
c              call cputim (n)
c      where n is a unique integer assigned to subroutine/process.
c
c-----
c      n          subroutine/process
c-----
c      11 . . . . read/initialization
c      12 . . . . stdyst
c      13 . . . . coefficients
c      14 . . . . solver
c      15 . . . . updtgem.f
c      16 . . . . kinrxns.f
c      17 . . . . output
c      18 . . . . path.f
c      18 . . . . trans.f
c      19 . . . . calcpsi.f
c      20 . . . . calcdpsi.f
c      21 . . . . factorization
c      22 . . . . gmres/bicgstab
c      23 . . . . nrtrans
c      24 . . . . react
c      29 . . . . total GEM cpu time
c      29 . . . . total GEM cpu time
c      30 . . . . cpu time per time step
c=====

      if(ipgm.le.0) then
        call seconds(tim1)
        if(ipgm.lt.0) then
          timmetra = tim1
          tingem = tim1
          do i = 1,30
            cpusub(i) = zero
            ncalls(i) = 0
          end do
        endif
        return
      else
        call seconds(tim2)
        sec = tim2-tim1

```

```

        if(ipgm.lt.28) then
            cpusub(ipgm) = cpusub(ipgm)+sec
            ncalls(ipgm) = ncalls(ipgm)+1
            return
        endif
    endif

    totsec =(tim2-timgem)

c    deltmin = deltmin*tcnvyr
c    deltmax = deltmax*tcnvyr

c    cpustep = (totsec-cpusub(1))/(mcyc+.1d0)
c    cputr   = (totsec-cpusub(1))/(itertot+.1d0)

    do i = 1,30
        percntg(i) = 100.d0*cpusub(i)/totsec
    end do

    write (*,9)
    if(ipgm.eq.28.or.ipgm.eq.30) then
        if(ipgm.eq.30) then
            write (iunit2,9)
            write (iunit0,9)
            write (iunit0,10) (cpusub(i),percntg(i),ncalls(i),i=1,7),
*                               cpusub(10),percntg(10),itime,nnewton,ncuts
        endif
        write (ifout,9)
        write (ifout,10) (cpusub(i),percntg(i),ncalls(i),i=1,7),
*                        cpusub(10),percntg(10),itime,nnewton,ncuts
        write (*, 10) (cpusub(i),percntg(i),ncalls(i),i=1,7),
*                    cpusub(10),percntg(10),itime,nnewton,ncuts
    else
        write (iunit2,9)
        write (iunit0,9)
    endif
    endif

    if(ipgm.ge.29) then
        ii = ihalcum
        write (iunit2,15) (cpusub(i),percntg(i),ncalls(i),i=11,25),
*                        mcyc,itertot,ihalcum
        write (iunit0,15) (cpusub(i),percntg(i),ncalls(i),i=11,25),
*                        mcyc,itertot,ihalcum
        write (*,15) (cpusub(i),percntg(i),ncalls(i),i=11,25),
*                    mcyc,itertot,ihalcum
    endif

    if(ipgm.eq.30) then
        write (iunit2,25) totsec
        write (ifout,25) totsec
        write (iunit0,25) totsec
        write (*,25) totsec
    endif

    write (iunit0,20)
    write (iunit2,20)
    write (ifout,20)

9    format(/'=====',
*          '=====')

```

```

*          '          Routines          cpu-seconds   % time   ',
*          '      Total Calls',/
*          '=====',
*          '=====')
10  format( ' Metra:  read/initialization      =',f10.2,f10.2,i12/
*          ' Metra:      pvt                  =',f10.2,f10.2,i12/
*          ' Metra:      pckr                 =',f10.2,f10.2,i12/
*          ' Metra:      accm-coefs            =',f10.2,f10.2,i12/
*          ' Metra:      solve                 =',f10.2,f10.2,i12/
*          ' Metra:      updt(psk)(vpk)        =',f10.2,f10.2,i12/
*          ' Metra:      output                =',f10.2,f10.2,i12/
*          ' Metra:      Total Metra Exec time =',f10.2,f10.2//
*          ' Metra: Total Number of Steps     =',i10,/
*          ' Metra: Total Newtonian Iters     =',i10,/
*          ' Metra: Total Time-Step Cuts      =',i10,/
*          '-----',
*          '-----')

15  format( ' GEM:    Read/Initialization      =',f10.2,f10.2,i12/
*          ' GEM:    Stdyst                    =',f10.2,f10.2,i12/
*          ' GEM:    coefficients              =',f10.2,f10.2,i12/
*          ' GEM:    solver                    =',f10.2,f10.2,i12/
*          ' GEM:    Update                    =',f10.2,f10.2,i12/
*          ' GEM:    kinrxns+ionex+monod       =',f10.2,f10.2,i12/
*          ' GEM:    outgem/graphs             =',f10.2,f10.2,i12/
*          ' GEM:    Transd                    =',f10.2,f10.2,i12/
*          ' GEM:    calcpsi                   =',f10.2,f10.2,i12/
*          ' GEM:    calcdpsi                  =',f10.2,f10.2,i12/
*          ' GEM:    factorization              =',f10.2,f10.2,i12/
*          ' GEM:    bicgstb/gmres              =',f10.2,f10.2,i12/
*          ' GEM:    nrtrans(x)(+solver)       =',f10.2,f10.2,i12/
*          ' GEM:    react                     =',f10.2,f10.2,i12/
*          ' GEM:    GEM execution time        =',f10.2,f10.2,i12//
*          ' GEM:    Number of Steps           =',i10,/
*          ' GEM:    Newtonian Iters           =',i10,/
*          ' GEM:    Time-Step Cuts            =',i10)

20  format('=====',
*          '=====')/
25  format('/' Total GEM + METRA CPU-Time      =',f10.2,'      100.')

      return
      end

c*deck seconds
c
      subroutine seconds (sec)

c =====
c
c      This subroutine returns cpu-seconds in variable 'sec'.
c      This routine MUST BE MODIFIED for a particular
c      operating system and the computer. If a suitable
c      system routine is not available, simply set 'sec'
c      = 0.d0 and 'return'.

c      Note that 'sec' is a double precision variable.

```

```

c =====
c      character*8 tim1,image
c      real*8 sec
c      dimension tyme(2)

c=====
csun      for sun  work station
csun      insert here the call for sun timing routine

c      sec = etime(tyme)
c      sec = tyme(1)
csun      sec = dtime(tyme)

c=====
c      for pc/lahey
cpc      call timer (isec)
cpc      sec = .01d0*isec

c=====
c      for pc/interactive-unix
c      call time (tim1)

c      write(image,10) tim1
c      read (image,20) ihr1,min1,isecl

c      sec = ihr1*3600+min1*60+isecl

c10      format(a8)
c20      format(3(i2,1x))
c      return
c=====

      end

c*datablock
c
c      block data metragem

c=====
c  This block data contains miscellaneous constants and arrays.
c=====
c
c      Date          Author(s)          Comments/modifications
c
c      Feb. 18, 95    Mohan S. Seth       Initial Implementation
c
c=====

      include 'metra/impl.h'
      include 'metra/frfmt.h'

      data izro,ione,itwo,ithr,ifour,ifive,isix,svn,ieight,inine,iten,
+      ififtn,itwnty/0,1,2,3,4,5,6,7,8,9,10,15,20/

      data zero,quarter,half,one,two,three,four,five,six,seven/
*      0.d0,0.25d0,0.5d0,1.d0,2.d0,3.d0,4.d0,5.d0,6.d0,7.d0/
      data eight,fnine,ten,onesixth,twthrds/
*      8.d0,9.d0,10.d0,.16666666666667d0,.66666666666667d0/

```

```
data ncol,nvalue,echo, echono,iblack, iplus,iminus/  
+ 163, 10, 'LIST', 'NOLI', ' ', '+', '-'/  
data igual,islash,istar,icmt, master, skip, noskip/  
+ '=', '/', '*', ':', 'MAST', 'SKIP', 'NOSK'/  
  
end
```



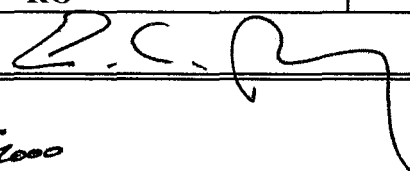
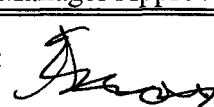
This file was created on: Fri May 19 11:31:20 2000

Copyright (c) 2000 Southwest Research Institute  
All Rights Reserved  
METRA test: Theis' solution for pumping from infinite aquifer

4-©	1 Sess-1	129.162.200.176	24/9
-----	----------	-----------------	------

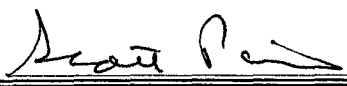
**MULTIFLO**  
**V.1.2.2**

SOFTWARE RELEASE NOTICE

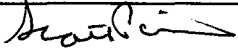

1. SRN Number: <b>PA-SRN-222</b>		
2. Project Title: <b>Near-Field Environment KTI</b>		Project No. <b>20-1402-561</b>
3. SRN Title: <b>MULTIFLO Version 1.2.2</b>		
4. Originator/Requestor: <b>Bruce Mabrito</b>		Date: <b>8/22/2000</b>
5. Summary of Actions		
<input type="checkbox"/> Release of new software		
<input checked="" type="checkbox"/> Release of modified software:		
<input type="checkbox"/> Enhancements made: DCM, (Unsturctured grid.)		
<input checked="" type="checkbox"/> Corrections made: Bugs corrected.		
<input type="checkbox"/> Change of access software		
<input type="checkbox"/> Software Retirement		
6. Persons Authorized Access		
Name	Read Only/Read-Write	Addition/Change/Delete
<b>Scott Painter</b>	<b>RW</b>	<b>Addition/Change/Delete</b>
<b>Debra Hughson</b>	<b>RO</b>	
<b>Lauren Browning</b>	<b>RO</b>	
<b>Ron Green</b>	<b>RO</b>	
<b>Goodluck Ofoegbu</b>	<b>RO</b>	
<b>Melissa Hill</b>	<b>RO</b>	
<b>Walter Illman</b>	<b>RO</b>	
<b>Mohan Seth</b>	<b>RW</b>	
<b>English Percy</b>	<b>RO</b>	
<b>Bret Leslie</b>	<b>RO</b>	
<b>Hans Arlt</b>	<b>RO</b>	
7. Element Manager Approval: 		Date: <b>8/22/2000</b>
8. Remarks:  <b>8/22/2000</b>		

2/9

# SOFTWARE SUMMARY FORM

01. Summary Date: <b>08/22/2000</b>		02. Summary prepared by (Name and phone) <b>Scott Painter, 522-3348</b>		03. Summary Action:  <b>New</b>
04. Software Date: <b>08/18/2000</b>		05. Short Title: <b>MULTIFLO Version 1.2.2</b>		
06. Software Title:  <b>MULTIFLO Version 1.2.2</b>				07. Internal Software ID:  <b>NONE</b>
08. Software Type:  <input type="checkbox"/> Automated Data System  <input checked="" type="checkbox"/> Computer Program  <input type="checkbox"/> Subroutine/Module		09. Processing Mode:  <input type="checkbox"/> Interactive  <input type="checkbox"/> Batch  <input checked="" type="checkbox"/> Combination		10. APPLICATION AREA a. General: <input checked="" type="checkbox"/> Scientific/Engineering <input checked="" type="checkbox"/> Auxiliary Analyses <input type="checkbox"/> Total System PA <input type="checkbox"/> Subsystem PA <input type="checkbox"/> Other  b. Specific: <b>Groundwater multiphase flow and reactive transport model</b>
11. Submitting Organization and Address: <b>CNWRA 6220 Culebra Road San Antonio, TX 78228</b>		12. Technical Contact(s) and Phone:  <b>Scott Painter, (210) 522-3348 Mohan Seth, (972) 699-3610</b>		
13. Narrative: <b>The code is used to model multiphase groundwater flow and reactive transport.</b>				
14. Computer Platform  <b>SUN</b>	15. Computer Operating System:  <b>UNIX</b>	16. Programming Language(s):  <b>Fortran 77</b>	17. Number of Source Program Statements:  <b>~80,000</b>	
18. Computer Memory Requirements: <b>Problem Dependent</b>	19. Tape Drives:  <b>N/A</b>	20. Disk/Drum Units:  <b>N/A</b>	21. Graphics:  <b>ASCII plot data files</b>	
22. Other Operational Requirements <b>Thermodynamic database required.</b>				
23. Software Availability:  <input checked="" type="checkbox"/> Available <input type="checkbox"/> Limited <input type="checkbox"/> In-House ONLY		24. Documentation Availability:  <input checked="" type="checkbox"/> Available <input type="checkbox"/> Inadequate <input type="checkbox"/> In-House ONLY <div style="text-align: right;"><b>DRAFT</b></div>		
<div style="display: flex; justify-content: space-between; align-items: flex-end;"> <div>Software Developer: </div> <div>Date: <b>8-22-00</b></div> </div>				

# SOFTWARE CHANGE REPORT (SCR)

<b>1. SCR No. (Software Developer Assigns):</b> <p>320</p>	<b>2. Software Title and Version:</b> <p>Multiflo V1.2.1</p>	<b>3. Project No:</b> <p>20-1402-562</p>
<b>4. Affected Software Module(s), Description of Problem(s):</b> <ol style="list-style-type: none"> <li>1. Water density calculation was using partial pressure instead of total pressure in (metra).</li> <li>2. Test for change from all liquid to 2-phase did not account for dissolved air (metra).</li> <li>3. Velocities in DCM unstructured runs were not printed correctly (metra).</li> <li>4. Air mole fraction at boundary was not calculated correctly with type 5 boundary condition.</li> </ol>		
<b>5. Change Requested by:</b> <p>Scott Painter  Date: August 17, 2000</p>	<b>6. Change Authorized by (Software Developer):</b> <p>Scott Painter  Date: August 17, 2000</p>	
<b>7. Description of Change(s) or Problem Resolution (If changes not implemented, please justify):</b> <ol style="list-style-type: none"> <li>1. Modified the following metra modules slightly to fix water density calculation:  pvt2o.f pvt.f pvtvp.f</li> <li>2. Updated the following metra modules phase change test: updtpsk.f updtpvk.f</li> <li>3. Fixed velocity printing problem in metra/plots.f</li> <li>4. Changed metra/bcond.f to set air mole fraction in incoming water with type 5 boundary condition</li> <li>5. Minor changes to metra/plots.f , gem/graph2d.f and gem/graph3d.f to fix printing of  TECPLOT header on output files.</li> <li>6. Small change to watsolv.f.</li> <li>7. Modified metra/iter.f to correct small problem with air balance.</li> <li>8 Updated version numbers on modified modules.</li> </ol>		
<b>8. Implemented by:</b>  <p>Scott Painter / Mohan Seth</p>	<b>Date:</b> August 17, 2000	
<b>9. Description of Acceptance Tests:</b> <p>Tested using the MULTIFLO standard test problem, as described in the Scientific Notebook 282 Volume 7, Entry dated 8.18.00. Input and output files are with the source code in the directory RUNS.</p>		
<b>10. Tested by:</b> Scott Painter 	<b>Date:</b> August 18, 2000	

4/9

**CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES  
DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE**

**DEVELOPED SOFTWARE<sup>1</sup>**

**Software Title/Name:** Multiflo V1.2.2

**Version:** V1.2.2

**Demonstration workstation:** Vulcan

**Operating System:** Solaris

**Developer:** S. Parmer

**1. Software Requirements Description: TOP-018, Section 5.3**

Software Requirements Description (SRD) and any changes thereto reviewed in accordance with QAP-002 requirements?

Yes: ☒ No: ☐ N/A: ☐

SRD Version: 1.2

SRD Approval Date: March 30, 1998

Notes:

**2. Software Development Plan (SDP): TOP-018, Section 5.4**

a) The Element Manager has approved the SDP and any changes?

Yes: ☐ No: ☐ N/A: ☒

b) The SDP addresses applicable section of TOP-018, Appendix B, Software Development Plan Template?

Yes: ☐ No: ☐ N/A: ☒

SDP Version:

SDP Approval Date:

Notes:

Software predates requirement for an SDP. See attached SCR# 320.

<sup>1</sup> See TOP-018, Table 1 for criteria.

5/9

**DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE  
DEVELOPED SOFTWARE**

**3. Design and Development: TOP-018, Section 5.5.1, 5.5.2**

- a) Is development and module/subroutine-level testing documented either in scientific notebooks and/or in Software Change Reports (SCR)?

Yes: ☒ No: ☐ N/A: ☐

Scientific Notebook(s): 282 Volume 7 Entry 8/18/00

SCR Number(s): 320

Notes:

- b) Is development and module/subroutine-level testing sufficiently documented so that an informed reviewer can follow the testing procedures and logic?

Yes: ☒ No: ☐ N/A: ☐

Notes: See Notebook # 282

- c) Is development in accordance with the conventions described in the SDP/SCR, i.e. coding convention?

Yes: ☒ No: ☐ N/A: ☐

Notes: Followed same style of existing code.

**4. Internal Documentation: TOP-018, Section 5.5.3**

Software internally documented to allow a user to understand the function(s) being performed and to follow the flow of execution of individual routines?

Yes: ☒ No: ☐ N/A: ☐

Module(s) Reviewed: updt psk.f  
updt vpk.f

Notes:

**5. Output: TOP-018, Section 5.5.4**

Software designed so that individual runs are uniquely identified by Date, Time, Name of software and version?

Yes: ☒ No: ☐ N/A: ☐

Date and time of run: August 17, 2000

Name and version: Metra v.1.2.2 / Multiflo v1.2.2

Notes: Metra is Part of Multiflo.

6/9

# DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE DEVELOPED SOFTWARE

## 6. Code Reviews: TOP-018, Section 5.5.5

Are code reviews (if implemented) documented in a scientific notebook or in another format that allows others to understand the code review process and results?

Yes: ☐ No: ☐ N/A: ☒

Scientific Notebook: \_\_\_\_\_

Notes: Acquired code that is not to be modified is accepted as is. No code reviews required. *No code reviews required*

## 7. Medium and Header Documentation: TOP-018, Section 5.5.6

a) Program title block of main program contains required information?

Yes: ☒ No: ☐ N/A: ☐

Program Title: Multiflo

Customer Name: NRC

Customer Office/Division: NRC 02-97-009

Customer Contact(s): Bret Leslie

Customer Phone Number: 301-415-6652

Associated Documentation: N/A

Disclaimer Notice: Yes

Notes:

b) Source code module header contains required information provides Program Name, Client Name, Contract Reference, Revision Number, and Revision History?

Yes: ☒ No: ☐ N/A: ☐

Module Reviewed: updtupk.f -OK

Module Reviewed: updt psk.f (No rev. history)\*

Module Reviewed: Main mlti.f -OK

Notes: \* Commented in body w/ revision/version number. Header corrected on Vulcan.



2/5

**DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE  
DEVELOPED SOFTWARE**

**7. Medium and Header Documentation, continued: TOP-018, Section 5.5.6**

- c) The physical labeling of software medium (tapes, disks, etc.) contain required information?

Yes: ☒ No: ☐ N/A: ☐

Program Name: Multiflo

Module/Name/Title: N/A

Module Revision: 1.2.2

File Type (ASCII, OBJ, EXE): Source code - ASCII

Recording Date: 8/21/00

Operating System of Supporting  
Hardware: Run on any Unix.

Notes:

**8. User's Manual: TOP-018, Section 5.5.5**

- a) Is there a Users' Manual for the software?

Yes: ☒ No: ☐ N/A: ☐

User's Manual Version and Date: V.2.1 V1.2 Feb. 2000

Notes: 8/21/00 Revision 2 Change 1

- b) Are there basic instructions for the use of the software?

Yes: ☒ No: ☐ N/A: ☐

Location of Instruction: User's Manual

Notes: Revision 2 Change 1

**9. Acceptance Testing: TOP-018, Section 5.6**

- a) Does the acceptance testing demonstrate whether or not requirements in the SRD and/or SCR have been fulfilled?

Yes: ☒ No: ☐ N/A: ☐

Location of Test Results: SCR 320

Notes: Refer to notebook 202

8/9

**DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE  
DEVELOPED SOFTWARE**

**9. Acceptance Testing, continued: TOP-018, Section 5.6**

- b) Has acceptance testing been conducted for each intended computer platform and operating system?

Yes: ☒ No: ☐ N/A: ☐

Platform(s): Sun & Intel

Operating System(s): Solaris & NT

Location of Test Results: Notebook 202, Vol 7, 8/18/00

Notes: NT test results on QA record copy of CD.  
Solaris test results on Vulcan.

- c) Has installation testing been conducted for each intended computer platform and operating system?

Yes: ☒ No: ☐ N/A: ☐

Platform(s): Sun & Intel

Operating System(s): Solaris & NT

Location of Test Results: Notebook 202, Vol. 7, 8/18/00

Notes: Done as part of acceptance testing

**10. Configuration Control: TOP-018, Section 5.7**

- a) Is the Software Summary Form completed and signed?

Yes: ☒ No: ☐ N/A: ☐

Software Summary Form Approval Date: 8/22/00

Notes:

- b) Is a software technical description prepared, documenting the essential mathematical and numerical basis?

Yes: ☒ No: ☐ N/A: ☐

Location Technical Description: User's Manual V1.2

Notes:

- c) Is the source code available (or, is the executable code available in the case of (acquired/commercial codes)?

Yes: ☒ No: ☐ N/A: ☐

Location of Source Code: QA record copy CD.

Notes: V. 1.2.2

9/9

**DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE  
DEVELOPED SOFTWARE**

**11. Configuration Control, continued: TOP-018, Section 5.7**

- d) Have all the script/make files and executable files been submitted to the Software Custodian?

Yes: ☒ No: ☐ N/A: ☐

Location of Script/Make Files: One in each directory

Notes: See record copy CD.

**12. Software Release: TOP-018, Section 5.9**

Upon acceptance of the software as verified above, has a Software release Notice, Form TOP-6 been issued?

Yes: ☐ No: ☐ N/A: ☐

Version number on software (1.0 for 1<sup>st</sup> issue): —

Version number on SRN: V 1.2.2

Notes: Approved 8/22/00

**13. Software Validation: TOP-018, Section 5.10**

- a) Has a Software Validation Test Plan (SVTP) been prepared for the range of application of the software?

Yes: ☐ No: ☐ N/A: ☒

Version/Date of SVTP: \_\_\_\_\_

Date reviewed and approved via QAP-002: \_\_\_\_\_

Notes: Not at this time

- b) Has a Software Validation Test Report (SVTR) been prepared that documents the results of the validation cases, interpretation of the results, and determination if the software has been validated?

Yes: ☐ No: ☐ N/A: ☒

Version/Date of SVTR: \_\_\_\_\_

Date reviewed and approved via QAP-002: \_\_\_\_\_

Notes: Not at this time.

Additional Remarks:

 8-22-00  
CNWRA Software Developer/Date

 8/22/00  
CNWRA Software Custodian/Date