DISTRIBUTION SHEET

To	From	Page 1 of 1
DISTRIBUTION	WATER RESOURCES ENGINEERING & MODELLING	Date December 29, 1994
Project Title/Work Order		EDT No. 123312
CERTIFICATION OF VERSION 1.2 OF THE PORFLO-3 CODE FOR THE WHC SCIENTIFIC AND COMPUTATIONAL CENTER		ECN No. N/A

Name	MSIN	Text With All Attach.	Text Only	Attach./ Appendix Only	EDT/ECN Only
L. L. Carter	H0-35	X			
L. B. Collard	H6-30	X			
M. P. Connelly	H4-85	X			
J. D. Davis	H0-33	X			
T. H. DeLorenzo	H0-36	X			
M. J. Fayer	K9-33	X			
E. J. Freeman	H0-36	X			
M. P. Johansen	A5-19	X			
R. Khaleel	H0-36	X			
N. W. Kline(15)	H0-36	X			
D. W. Langford	B5-25	X			
A. H. Lu	H0-36	X			
F. M. Mann	H0-36	X			
M. G. Piepho	H0-36	X			
J. A. Rawlins	H0-36	X			
P. D. Rittmann	H6-30	X			
D. R. Sherwood	B5-01	X			
J. C. Sonnichsen	H6-06	X			
R. K. Stewart	H4-83	X			
Central Files(2)	L8-04	X			
OSTI (2)	L8-07	X			
B. W. Drost, U.S. Geological Survey 1201 Pacific Avenue, Suite 600 Tacoma, WA 98402		X			
C. Cline, Washington State Department of Ecology P. O. Box 47600 Olympia, WA 98504-7600		X			
M. Th. van Genuchten, U.S. Salinity Laboratory 4500 Glenwood Drive Riverside, CA 92501		X			
A. K. Runchal Analytic and Computational Research, Inc. 1931 Stradella Road Los Angeles, CA 90077		X			

WHC-SD-WM-CSWD-069
Revision 0

Certification of Version 1.2 of the PORFLO-3 Code for the WHC Scientific and Engineering Computational Center

Prepared for the U.S. Department of Energy
Office of Environmental Restoration and
Waste Management

Westinghouse
Hanford Company Richland, Washington

Hanford Operations and Engineering Contractor for the
U.S. Department of Energy under Contract DE-AC06-87RL10930

Approved for Public Release

DISCLAIMER

549.21 JAN 10 1995

ENGINEERING DATA TRANSMITTAL

1. EDT 123312

2. To: (Receiving Organization): Distribution

3. From: (Originating Organization): Water Resources Engineering and Modelling

4. Related EDT No: 123307

5. Proj/Prog/Dept/Div: Liquid Effluents

6. Cog/Proj Engr: NW Kline

7. Purchase Order No: NA

8. Originator Remarks:

Supporting Document Approval Requested. Comments welcomed.

9. Equip/Component No: NA

10. System/Bldg/Facility: NA

11. Receiver Remarks:

12. Major Assm Dwg No: NA

13. Permit/Permit Application No. NA

14. Required Response Date: 9/9/94

15. DATA TRANSMITTED

(A) Item No.	(B) Document/Drawing No.	(C) Sheet No.	(D) Rev No.	(E) Title or Description of Data Transmitted	(F) Impact Level	(G) Reason for Transmittal	(H) Originator Disposition	(I) Receiver Disposition
1	WHC-SD-WM-CSWD-069		0	Certification of Version 1.2 of the PORFLO-3 Code for the WHC Scientific & Engineering Computational Center	SQ	1	1	

16. KEY

Impact Level (F)	Reason for Transmittal (G)	Disposition (H) & (I)
1, 2, 3, or 4 see MRP 5.43 and EP-1.7	1. Approval 2. Release 3. Information 4. Review 5. Post-Review 6. Dist (Receipt Acknow. Required)	1. Approved 2. Approved w/comment 3. Disapproved w/comment 4. Reviewed no/comment 5. Reviewed w/comment 6. Receipt acknowledged

17. SIGNATURE/DISTRIBUTION

(See Impact Level for required signatures)

(G) Reason	(H) Disp	(J) Name	(K) Signature	(L) Date	(M) MSIN	(J) Name	(K) Signature	(L) Date	(M) MSIN	(G) Reason	(H) Disp
1	1	Cog./Proj. Eng NW Kline	[signature]	8/25/94	HO-36						
1	1	Cog./Proj. Eng. Mgr. JA Rawlins	[signature]	8/25/94	HO-36						
1	1	QA ML Hermanson	[signature]	11-17-94	A4-79						
1	1	Safety JC VanKeuren	[signature]	12/1/94	H4-64						
3		LL Carter	[signature]	8-25-94							

18. [signature] 8/25/94
Signature of EDT Originator — Date

19. Authorized Representative for Receiving Organization — Date

20. [signature] 8/25/94
Cognizant/Project Engineer's Manager — Date

21. DOE APPROVAL (if required)
Ltr No.
☐ Approved
☐ Approved w/comments
☐ Disapproved w/comments

RELEASE AUTHORIZATION

Document Number:	WHC-SD-WM-CSWD-069, REV.0
Document Title:	Certification of Version 1.2 of the PORFLO-3 Code for the WHC Scientific & Engineering Computational Center
Release Date:	12/29/94

This document was reviewed following the procedures described in WHC-CM-3-4 and is:

APPROVED FOR PUBLIC RELEASE

WHC Information Release Administration Specialist:

Niki Solis 12/29/94

Niki Solis

SUPPORTING DOCUMENT

1. Total Pages 28

2. Title	3. Number	4. Rev No.
Certification of Version 1.2 of the PORFLO-3 Code for the WHC Scientific and Engineering Computational Center	WHC-SD-WM-CSWD-069	0

5. Key Words

Numerical Modelling, Acceptance Testing, Workstation

6. Author

Name: NW Kline

Signature 8/10/94

Organization/Charge Code 86910/J812A

7. Abstract 12/29/94 N.S.

Version 1.2 of the PORFLO-3 Code has migrated from the Hanford Cray computer to workstations in the WHC Scientific and Engineering Computational Center. The workstation-based configuration and acceptance testing are inherited from the CRAY-based configuration. The purpose of this report is to document differences in the new configuration as compared to the parent Cray configuration, and summarize some of the acceptance test results which have shown that the migrated code is functioning correctly in the new environment.

9. Impact Level SQ

10. RELEASE STAMP

OFFICIAL RELEASE BY WHC
DATE JAN 10 1995
Sta. 21

CERTIFICATION OF VERSION 1.2 OF THE PORFLO-3 CODE FOR THE WHC SCIENTIFIC AND ENGINEERING COMPUTATIONAL CENTER

Prepared by: [signature] 6/30/94
N. W. Kline
Alternate Code Custodian

Reviewed by: David Warren Langford 8/11/94
D. W. Langford
Code Custodian

Approved by: John A. Rawlins 8/18/94
J. A. Rawlins, Manager
Environmental Risk and
Performance Assessment

TABLE OF CONTENTS

List of Tables

List of Figures

APPENDICES

ACKNOWLEDGEMENT OF TRADEMARKS

ACRi, **PORFLOW** and **PORFLO-3** are trademarks of Analytic and Computational Research, Inc.

CRAY is a registered trademark of Cray Research, Inc.

IBM and **RISC System/6000** are registered trademarks of International Business Machines Corporation.

Silicon Graphics is a trademark of Silicon Graphics, Inc.

SPARCstation is a trademark of SPARC International, Inc., licensed exclusively to SUN Microsystems, Inc.

SUN Microsystems and **SUN Workstation** are registered trademarks of SUN Microsystems, Inc.

TECPLOT and **PREPLOT** are registered trademarks of Amtec Engineering, Inc.

UNIX is a registered trademark of AT&T.

1.0 INTRODUCTION

1.1 BACKGROUND

Version 1.2 of the PORFLO-3 computer code (Runchal, et al., 1992) was recently certified by Westinghouse Hanford Company (WHC) as being suitable for use at "Impact Level 2" (Kline, 1993). That initial certification was specific to the Hanford CRAY computer which was then the scientific and engineering computational platform of choice at Hanford, but which is now retired. Completion of the initial certification effort included not only documenting the CRAY software configuration of Version 1.2 of the PORFLO-3 code, but also substantial acceptance testing based on verification and benchmark testing of an earlier code version.

In anticipation of retirement of the Hanford CRAY system, Version 1.2 of PORFLO-3 was migrated to UNIX workstations which are now nodes in the new WHC Scientific and Engineering Computational Center (SECC). Some of those same workstations were previously used for postprocessing results of PORFLO-3 simulations excuted on the Hanford CRAY. Now some compute platforms of the SECC are also targeted for execution of PORFLO-3. Acceptance testing in the new configuration is based on that which has already been documented for Version 1.2 of PORFLO-3 (Kline, 1993).

1.2 PURPOSE

The workstation based configuration of Version 1.2 of the PORFLO-3 code, the process of building executables for the workstations, and acceptance testing of the new executables are described herein. The purpose is to substantiate the claim that Version 1.2 of PORFLO-3 is suitable for use on targeted workstations in the SECC. Although the "Impact Level" designation and terminology used in the previous certification documentation is now obsolete, the standards of quality are the same and the same quality assurance and administrative approval is sought hereupon.

1.3 SCOPE

The workstation based PORFLO-3 configuration, the process of building executables, and acceptance testing thereof are inherited, and are in large part identical to CRAY based counterparts which have been documented and approved. The scope of the present effort is focused on identification of differences between the workstation implementation and the documented CRAY implementation.

2.0 WORKSTATION SOFTWARE CONFIGURATION

2.1 VERSION NUMBER, RELEASE DATE, AND LICENSE NUMBER

The 12AUG92 Release of Version 1.2 of the PORFLO-3 code was marked by the issuance of the users guide (Runchal et al., 1992). The August 12, 1992 release date is the date on which both the source code was completed for the CRAY master cycle (License Number 3120WHC100) and the example problems in Appendix B of the users guide were completed. The exact same source code is still being used today; i.e., to date the source code file for the master cycle of Version 1.2 of PORFLO-3, p3d.src, is unchanged. Moreover, the source for the associated ITPACK solvers, texas.src, is also unchanged. Hence, although the Hanford CRAY computer is now retired, the CRAY configuration could be re-established on another CRAY computer elsewhere, if needed, with a minimum of effort.

In 1993, the source for the master cycle (p3d.src and texas.src) was migrated to workstations in what is now the WHC SECC. The SECC is a collection of over 30 UNIX-based workstations along with peripherals, with access thru the same networks that also provided access to the Hanford CRAY computer. Target compute platforms for PORFLO-3 presently include all SUN Workstations, all Silicon Graphics (SGI) workstations, and all IBM RISC System/6000 workstations in the SECC. Base cycle numbering for the purpose of construction of executables is in accordance with that proposed previously (Kline, 1993), and is shown below.

Platform	Base Cycle
SGI	200
SUN	300
IBM	400

License numbering of executable cycles is also unchanged, but with an added constraint on the cycle number. The first digit of the cycle number is the platform indicator consistent with the base cycle numbers shown above. The last two digits of the three-digit cycle number indicate a unique combination of adjustable PARAMETERs. Therefore there can be at most 100 distinct combinations of the adjustable parameters (numbered 0 to 99) under the present cycle numbering scheme. This should be sufficient, based upon historical needs for executable cycles. The combinations of adjustable parameters previously and presently in use are shown in Table 1. Under the present cycle number interpretation, the last two digits of the cycle number are now referred to as the parameter set identifier.

Table 1. Parameter Sets for Cycles of PORFLO-3, Version 1.2

Date	Parameter Set Identifier	Parameters										Application
		LX	LY	LZ	LMX	LSS	LSO	LUS	LZN	LTRA	MXSTP	
8/12/92	00	100	4	100	100	10	100	100	10	10	32K	Users Guide & Tank Leak
8/20/92	01	100	10	100	100	10	100	100	10	10	60K	Tank Leak
9/23/92	02	40	40	40	40	10	100	100	10	10	60K	200-BP-1 OU
12/9/92	03	60	90	30	90	10	100	100	10	10	60K	300-FF-5 OU
1/13/93	04	10	10	110	110	10	100	100	10	10	60K	Test Problem VT-1
1/13/93	05	40	40	40	40	10	100	100	10	10	200K	200-BP-1 OU
2/10/93	06	70	4	70	70	10	100	100	20	10	100K	216-U-17 Crib
2/10/93	07	70	90	30	90	10	100	100	10	10	50K	300-FF-5-OU
3/25/93	08	053	0011	052	053	10	100	100	20	10	32K	ACRI, USDA
5/20/93	09	60	4	140	140	10	100	100	20	10	50K	Tank Leak II
6/10/93	10	60	110	30	110	10	100	100	10	10	50K	300-FF-5-OU
7/1/93	11	110	40	30	110	10	100	100	10	10	50K	Test Problems VT-3 & VT-4
10/13/93	21	110	10	10	110	10	100	100	10	10	60K	Test Problem VT-3
10/13/93	22	80	40	40	80	10	100	100	10	10	60K	Test Problem VT-4
10/14/93	23	4	40	50	50	10	100	100	20	10	100K	216-U-14 Ditch
10/14/93	24	40	40	50	50	10	100	100	20	10	100K	216-U-14 Ditch
10/22/93	25	4	80	60	80	10	100	100	20	10	100K	216-U-14 Ditch

As an example, cycle number 210 for the SGI workstations must have the same parameterization as cycle 110 did for the Hanford CRAY; however, existence of CRAY cycle 110 does not preclude existence of SGI cycle 210. Given a unique combination of parameters, it might be that executables do not exist with that parameterization for all workstations. A checklist of executable cycles presently available is shown in Table 2. It is intended that the Code Custodian will issue an Engineering Change Notice (ECN) periodically (at least annually) to provide updated copies of Tables 1 and 2, unless there are no changes to report.

2.2 CONSTRUCTION OF EXECUTABLES

Although the source code files for the CRAY master cycle (p3d.src and texas.src) are used to build executables for the targeted workstations and the license numbering of executables is unchanged, the command sequence for construction of an executable on any of the workstations is substantially more involved than it was for the CRAY. In the CRAY environment, a copy of p3d.src was edited to make appropriate PARAMETER changes and the executable was then constructed with three comands - generically, update, compile, and load (cf. Table 3 from Kline, 1993). Those first two steps (update and compile) are more involved in the workstation environment. First, a copy of p3d.src is used for the process of building an executable for a workstation, but it is <u>not</u> edited. An update patch file is used to apply source code corrections for compatibility with FORTRAN77 on the workstations, and <u>also</u> inserts PARAMETER changes and the cycle number. A standard patch file is maintained by the Code Custodian, and a copy of <u>it</u> must be edited prior to constructing an executable for the purpose of changing parameterization and related cycle numbering. A separate patch file is used to apply some limited syntax corrections to the source for the ITPACK solvers (texas.src) for compatibility with FORTRAN77 compilers on the workstations, but is never edited.

The UPDATE processor that is generally available in CRAY operating environments is <u>not</u> generally available in the workstation environment. A substitute update processor, called PRPR, has been adapted for use in application of patch files to the PORFLO-3 sources (p3d.src and texas.src). The PRPR code is itself a FORTRAN77 program, developed by the Los Alamos National Laboratory (LANL) developers of the MCNP code. The WHC adaptation of PRPR for use with PORFLO-3 is called PRPRW and is described in more detail in Appendix A. Since PRPRW is not part of the workstation operating environment, it must as well be maintained by the Code Custodian.

Table 2. Current Executable Cycles of PORFLO-3, Version 1.2

Parameter Set Identifier	SGI			SUN			IBM			Remarks
	Cycle	Date	Size (MB)	Cycle	Date	Size (MB)	Cycle	Date	Size (MB)	
00	200	10/13/94	14.3	300	10/13/93	14.3	400	10/14/93	15.8	
01										
02										
03										
04	204	10/13/93	4.8	304	10/13/93	4.8	404	10/14/93	5.0	
05										
06										
07										
08										
09										
10										
11										
21	221	10/13/93	4.8	321	10/13/93	4.8	421	10/14/93	5.0	
22	222	10/13/93	52.9	322	10/13/93	52.9	422	10/14/93	59.9	ibm1 & ibm2 each have 256 MB of primary memory
23							423	10/14/93	3.5	sgi1 & sgi2 each have 256 MB of primary memory
24							424	10/14/93	3.8	
25							425	10/22/93	7.8	erpa indigo machines have 90 MB
										SUN machines have
										48-64 MB

Note: Size of executable in MegaBytes is determined with the UNIX "size" command. Actual average memory requred at runtime might be different (typically less).

Once the update sources have been "patched" with PRPRW, FORTRAN77 compilation is performed on a modular basis. This was required initially in the migration effort due to physical/logical limits on the size of the working disk partition used by the FORTRAN77 compiler in the SUN environment, and is presently still the standard practice. Furthermore, the individual command line for compiling one FORTRAN77 source module varies between workstations. For instance, the compilation command line for a PORFLO-3 module named "_____.f" on SUN or SGI is

f77 -Bstatic -O2 -r8 -i4 -w ____.f ,

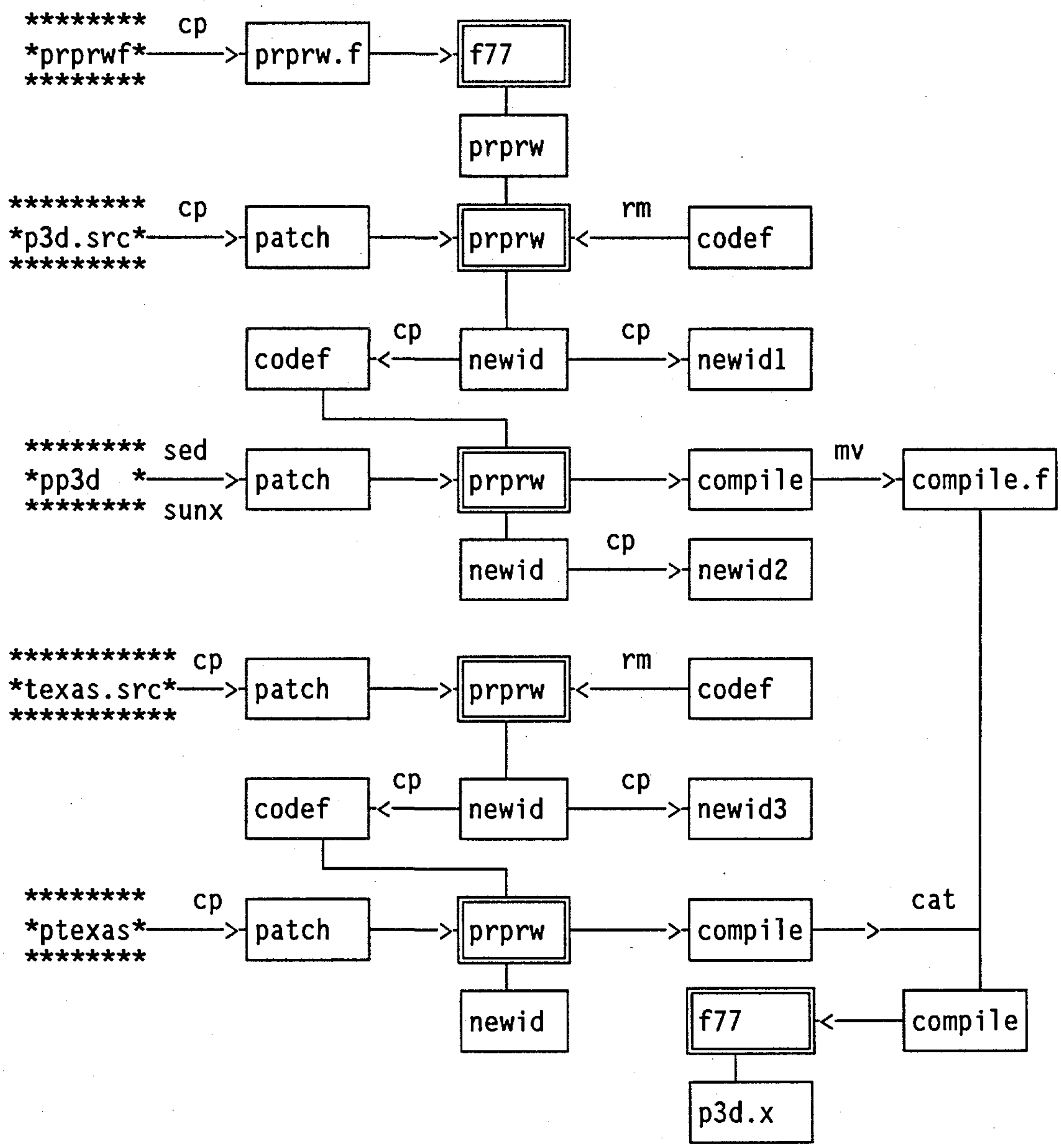
whereas the IBM RISC counterpart command line is

f77 -qsave -O2 -qautodbl=dblpad ____.f .

Loading the resulting object files into an executable is invoked by the compiler. Along with the FORTRAN77 object files, a file named r_timdat.o must be present in the working directory. The r_timdat.o file contains relocateable C code which interfaces to workstation equivalents of CRAY specific routines which in turn fetch the current date, time of day, and CPU time remaining for a batch job.

The full set of commands for updating, compiling and loading an executable for SGI, SUN or IBM workstations respectively is contained in UNIX script file ucl.sgi, ucl.sun or ucl.ibm, respectively. As an example, a copy of the present ucl.ibm script for creating an executable on an IBM RISC system is shown in Appendix B. A copy of the standard patch file tailored for creation of executable cycle number 200 (for the SGI environment) is shown in Appendix C. A summary of the logic built into the ucl.___ scripts is shown in Figure 1. Note that Figure 1 is logically generic to all workstations, but is specific to SUN platforms in particular only in the respect that the sed editor is shown changing sgix to sunx. Referring to the patch file shown in Appendix B, sgix, sunx, and ibmx respectively determine the platform indicator digit of the cycle number for SGI, SUN, and IBM platforms, respectively.

Figure 1. Flow Diagram for Obtaining PORFLO-3 Executable, p3d.x, on Sun (with command file ucl.sun)



---

mnemonic:

[xxxx]	file name xxxx	cp -- copy file mv -- move file cat -- concatenate files
******* *xxxx * *******	input file xxxx from disk	sed ——> sunx: use sed editor to change sgix to sunx
[[xxxx]]	execute file xxxx	

3.0 INTERFACES

Interfaces from Version 1.2 of PORFLO-3 to workstation-based visualization software have not evolved beyond the capabilities described previously (Kline, 1993). Version 1.2 of MKHIS and Version 1.2 of CMAX are still compatible with Version 1.2 of PORFLO-3. The source code files for MKHIS and CMAX, mkhis.f and cmax.f respectively, have been edited minimally to make the source code compatible with FORTRAN77 compilers on <u>all</u> SGI, SUN and IBM RISC workstations in the SECC. The source code file for Version 1.2 of MKTEC, mktec.f, has also been edited similarly; moreover, a minor deficiency in MKTEC has also been addressed. It was always intended that Version 1.2 of MKTEC would label each TECPLOT "zone" with the simulation time of the data in the zone. That feature was not functional at the time of the previous documentation (Kline, 1993), but is now operational in Version 1.2 of MKTEC.

An interface executable for MKTEC, MKHIS, or CMAX respectively is created in SGI and SUN environments using the command

```
f77 -Bstatic -O2 -r8 -i4 -w -o ____.x ____.f
```

where "____" is mktec, mkhis, or cmax, respectively. In the IBM environment, an executable is created from

```
f77 -qsave -O2 -qautodbl=dblpad -o ____.x ____.f  .
```

Input and output file naming conventions and data flow shown in Figure 2 by Kline (1993) are unchanged. It is noteworthy that TECPLOT 6.0 is now installed on the workstations along with the compatible PREPLOT, and that they are fully upward compatible; i.e., no changes were necessitated in the subject preprocessors due to upgrading TECPLOT. However, even though TECPLOT 6.0 with 3D volume features is installed, the current version of MKTEC still only produces 2D slices parallel to coordinate planes. Therefore the 3D volume features of TECPLOT 6.0 are <u>not</u> presently accessible through Version 1.2 of MKTEC.

4.0 FILE STORAGE AND ACCESS CONTROL

Master copies of all files needed in the construcion of executables for Version 1.2 of PORFLO-3 and compatible interfaces on the targeted workstations as described in the preceeding sections are safeguarded by the Code Custodian in the Hanford Common File System (CFS). The Code Custodian also controls storage of the executables indicated in Table 2 as well as all input and output files from the test problems indicated in the following section. Independent copies of all essential files are also maintained by the Code Custodian on separate media as a backup in the unlikely event that CFS and/or the SECC are/is unavailable for a prolonged period of time. Other secondary backup of essential files is maintained by the Alternate Code Custodian. Structure and node naming conventions in the storage directories are similar to those established previously [e.g. Figure 3 by Kline (1993)], and are not reiterated here.

Access to all master files is restricted to the Code Custodian. Access to executables is provided by allowing universal execute access to copies of executables placed in public areas of workstation disks. In the previous CRAY implementation, copies of executables were made available to users to facilitate independence of the users from the custodian. There was no effective means of transporting the executables from that system, and so egress control was also reasonably well-assured. Provision of execute-only access to executables on the workstations provides a similar level of egress control, but is implemented primarily to minimize the number of disk-resident executable files and thereby maximize available disk storage space. Other access to files must presently be arranged with the custodian on an individual basis.

Executable cycles of Version 1.2 of PORFLO-3 identified for Silicon Graphics platforms in Table 2 along with compatible executables for the graphics postprocessing interfaces are stored in directory /p_sgi/porflo. Similarly, SUN and IBM executables respectively are stored in directories /p_sun/porflo and /p_ibm/porflo, respectively. Each PORFLO-3 executable file stored in any of those directories has a file name of the form p3dnnn.x, where the string 'nnn' is the cycle number. With the parameter set identifier (last two digits of the cycle number), one can find the respective parameterization from Table 1. The executable size in megabytes is shown in Table 2. Executable files for MKTEC, MKHIS, and CMAX respectively have file names of the form

____.x, where '____' is mktec, mkhis, or cmax, respectively.

5.0 ACCEPTANCE TESTING

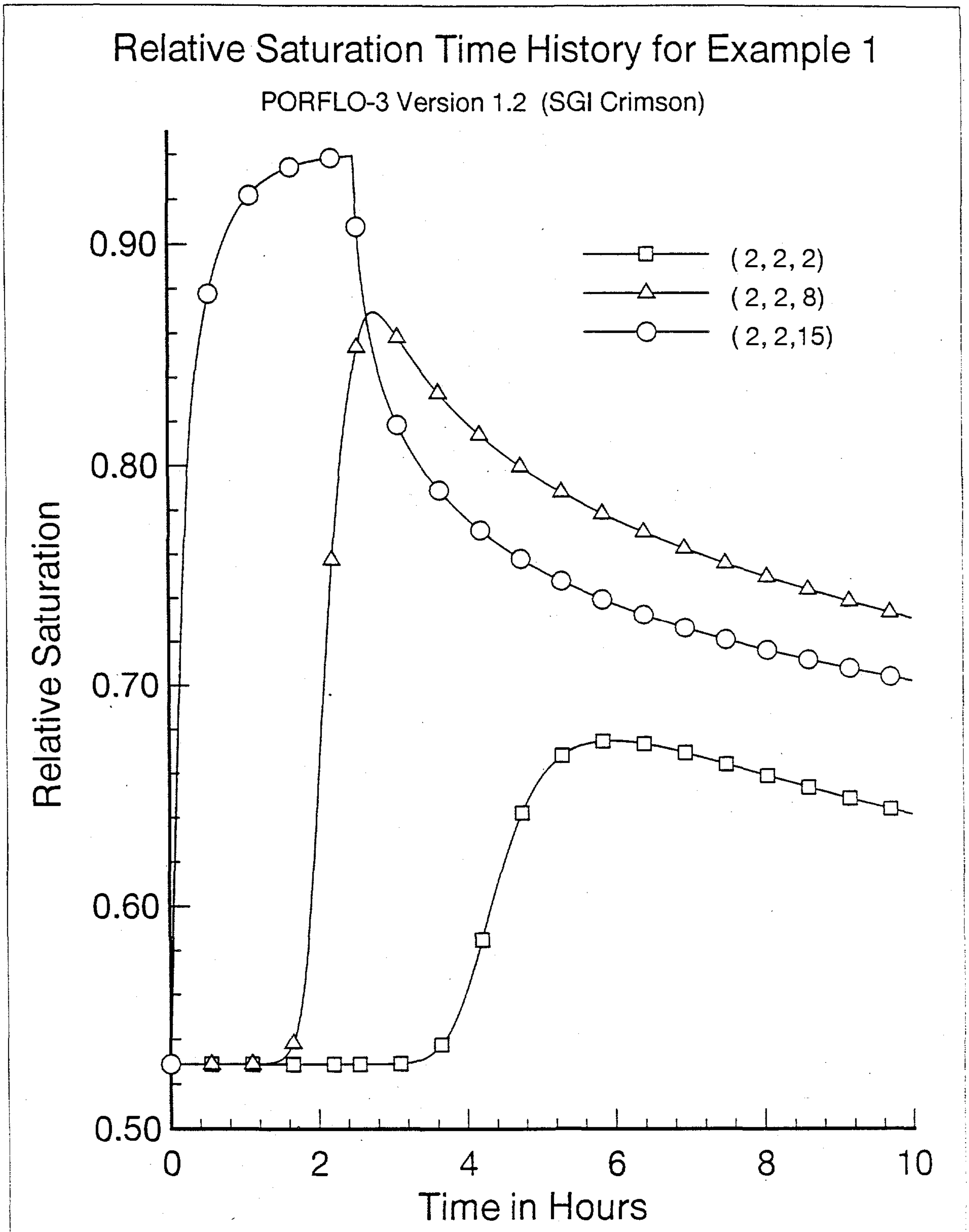
Acceptance testing with Version 1.2 of PORFLO-3 in the workstation-based configuration was conducted to demonstrate that the code is installed properly and functioning correctly. The basis for this testing was the set of test problems which were standardized with the parent CRAY-based configuration (Kline, 1993). In particular, this testing makes use of those test problems in final form (i.e., test problems VT-1, VT-2, VT-3c, VT-3p, VT-4c, BT-1, BT-2 and BT-5). Each was executed with PORFLO-3 on a SUN workstation, a Silicon Graphics workstation, and an IBM workstation. The first acceptance criterion for this exercise is that each test problem must execute successfully and terminate normally on each type of platform. The second criterion is that each problem must repeat the results obtained previously (Kline, 1993), without inexplicable exceptions. Automated comparison of coded output files from the latest test executions to those from the previous CRAY test executions was used as the basis for assessing relative goodness of test results under the second criterion. Success or failure according to the first criterion is measured by simply observing the process of execution of the test problems. The third acceptance criterion is that graphic results are also replicable.

Implicit in this last criterion is proper installation and functional correctness of the visualization interface postprocessors (MKTEC, MKHIS and CMAX).

Results of the tests are voluminous and are not recreated here; however, the Code Custodian maintains all input and output files for each test problem from each type of compute platform, and also maintains a log of the test executions and records of the comparisons of results to the CRAY-based standards. Of importance here is that each test problem was successful on each platform, with respect to all three acceptance criteria. Hence Version 1.2 of PORFLO-3 and compatible postprocessor interfaces are properly installed and functioning correctly on the targeted platforms of the SECC. A sample of graphic results is shown in Figures 2 and 3 to coroborate the conclusion. Figure 2 shows the time history of relative saturation from Example 1 as simulated on a Silicon Graphics Crimson workstation, and is comparable to Figure B-3 from Runchal et al. (1992). Figure 3 shows concentration contours for test problem BT-5 in the y-z plane at x = 64 meters after 100 days simulated time, and is comparable to Figure 18 by Kline (1993). Neither Figure 2 nor Figure 3 alone is conclusive; each is a summary of the end results of an appreciable sequence of computations that, to the human eye, appears to have repeated the same sequence of computations executed previously in the CRAY-based configuration. Preponderence of the full set of the Code Custodian's test results fully supports that observation.

A summary matrix of the eight test problems is contained in Table 3. The table shows the cycle number and cpu time for each problem on each platform. It also shows the percentage difference between workstation cpu time and CRAY cpu time for each test problem and each platform type. (A minus percentage indicates that the workstation executed in <u>less</u> cpu time than the CRAY.) The CRAY cpu times are from Table 4 by Kline (1993), and were produced on the Hanford CRAY X-MP/232 EA which featured an 8.5 nanosecond clock cycle. For completeness, the SGI workstation used for the present acceptance testing is a Silicon Graphics Crimson R4000, the SUN workstation is a SUN SPARCstation 10, and the IBM workstation is an IBM RISC System/6000 Model 580. The SGI and SUN workstations operate in the range of about 50 to 70 megahertz (from about 14 to 20 nanoseconds per cycle), i.e., about half the speed of the the CRAY cpu; the IBM workstation is comparable to the CRAY in processing speed. These facts generally correlate to the timing statistics shown in Table 3. Although throughput from any single workstation in the SECC might not compare to that experienced previously with the Hanford CRAY, turnaround times for long-running batch jobs run on workstations in the SECC can be quicker than CRAY counterparts since the workstation batch queueing convention is sequential processing of batch jobs.

Figure 2. Time History of Relative Saturation for Example Prolem 1, SGI Crimson.



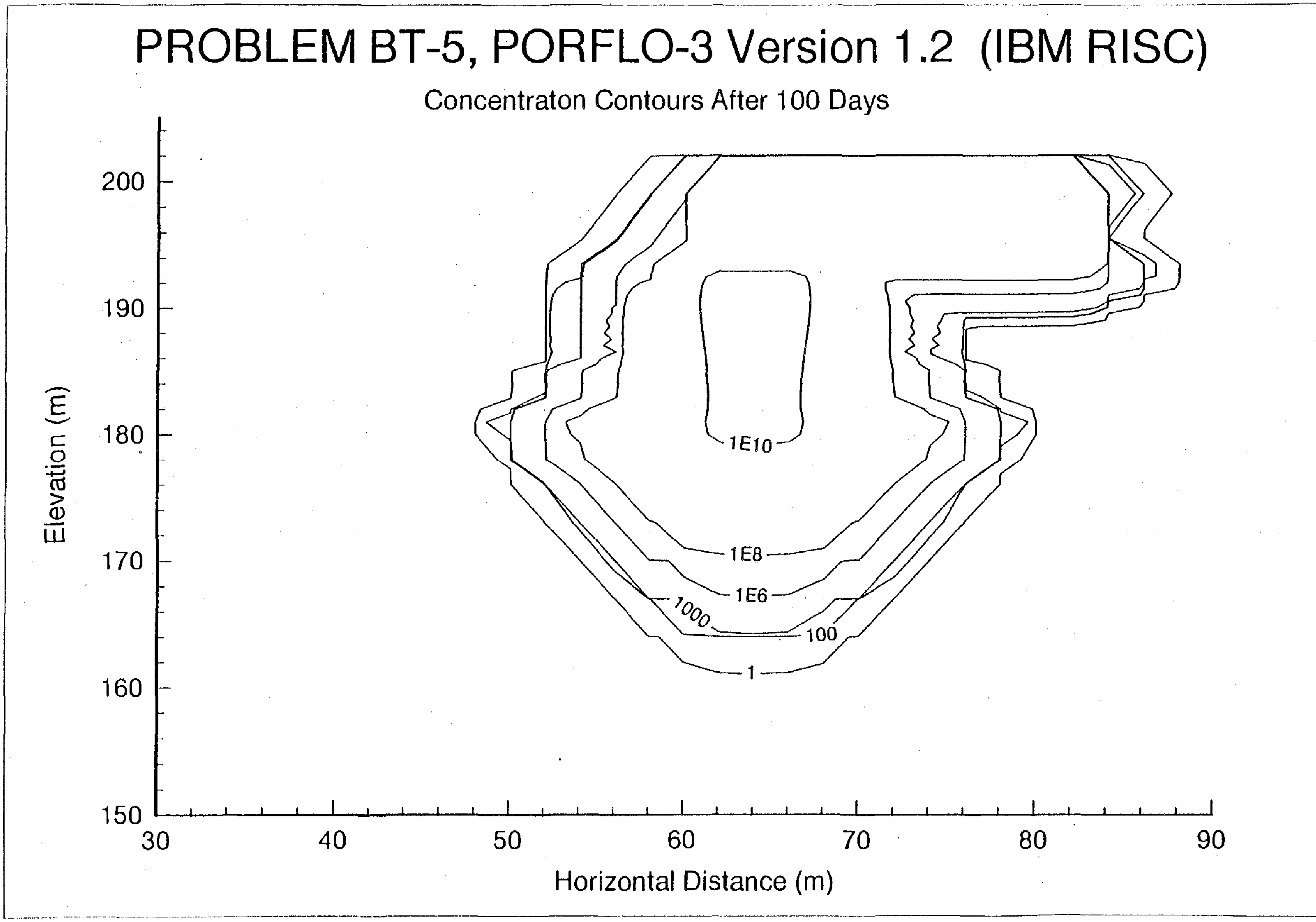


Figure 3. Concentration Contours for Test Problem BT-5, IBM RISC.

Table 3. Summary of PORFLO-3 Test Problem Runs

Problem Number	SGI				SUN				IBM				Remarks
	Cycle Number	CPU Time (Secs)	*% CRAY	Wall Clock (Secs)	Cycle Number	CPU Time (Secs)	% CRAY	Wall Clock (Secs)	Cycle Number	CPU Time (Secs)	% CRAU	Wall Clock (Secs)	
VT-1	204	40.8	50%	41.00	304	61.4	126%	107.00	404	23.6	-13%	70.7	
VT-2	200	13.0	46%	13.60	300	18.6	108%	32.63	400	5.7	-36%	6.6	
VT-3c	221	20.2	62%	20.30	321	28.0	124%	50.15	421	8.3	-34%	24.7	
VT-3p	221	11.6	99%	11.64	321	15.9	173%	16.12	421	4.4	-24%	12.8	
VT-4c	222	1955.3	220%	1988.4	322	2551.5	318%	2640.0	422	726.1	19%	2148.0	20K Bytes actual memory on ibm1
BT-1	200	112.8	72%	118.20	300	154.7	136%	241.30	400	51.6	-21%	53.1	
BT-2	200	1184.4	29%	1212.2	300	1764.2	92%	2700.8	400	554.9	-39%	572.0	
BT-5	222	4738.2	94%	4959.0	322	6940.8	184%	7257.0	422	2031.9	-17%	2068.0	
Example1	200	7.43		8.13									
Example2	200	2813.3		2872.0									

* % Cray = (Workstation cpu time/Cray cpu time -1) x 100% is the precentage difference from the original Cray cpu times.

6.0 CONCLUSIONS

Version 1.2 of PORFLO-3 is installed for general use on workstations in the WHC SECC, and is functioning correctly by virtue of acceptance testing based on previous verification and benchmarking tests. Realizations of cpu time for PORFLO-3 test cases in the SECC are generally greater for most targeted workstations than on the Hanford CRAY, but wall-clock turnaround times can be competitive for batch jobs. Graphic evidence indicates that not only is PORFLO-3 properly installed and functioning correctly, but so also are the interfaces to visualization software.

7.0 RECOMMENDATIONS

Some of the recommendations from the previous certification effort (Kline, 1993) have already been incorporated into the present work; i.e., test results from the CRAY are the basis for acceptance testing on workstations, cycle numbering follows the proposed numbering scheme, Example Problem 1 from the users guide is always used as the absolute minimum acceptance test for any new executable cycle, and the local TECPLOT installation has been upgraded to Version 6.0. The following are in part remaining for some further consideration and/or are now more clear in light of the new configuration and recent applications.

- Requirements for MKTEC capabilities need to be firmly established. Capability for the user to specify a 3D subdomain as a zone for use with the 3D volume feature of TECPLOT is a principal requirement. Some optional zone naming capability is also desirable. Other requirements might include having a single TECPLOT interface which is possibly not only compatible with all present versions of PORFLO-3/PORFLOW but also with other codes such as VAM3D, TEMPEST, etc. This involves at least some minimal software development.

- Although there is some incentive to adopt a single version of PORFLO-3 or PORFLOW in the interest of minimizing costs associated with software maintenance and training, each is unique much the same as PORFLOW and VAM3D are distinctively different and both are familiar. The cost of abandoning either for a newer version of <u>some</u> code which is constantly evolving and which is not as mature could be higher in terms of maintenance and retraining. Therefore each should be retained as long as each effectively satisfies a need. At the same time, it is also worthwhile to keep abreast of new code developments, particularly PORFLOW Versions 2.5 and 3.x which also have some unique features.

- A long-term arrangement with Analytic and Computational Research, Inc. (ACRi) for offsite backup and interface/distribution to other government agencies is still recommended, and could also facilitate the process of merging all of the best/necessary features of known versions of the PORFLO-3/PORFLOW codes into a single version which could become the site standard.

- As long as Version 1.2 of the PORFLO-3 code is being used, the Code Custodian will continue to gradually assume a more active role in not only maintaining files, building executables and controlling access but also in keeping the documentation up to date. At a minimum, an ECN should be issued annually to update Table 2 (unless of course there are no changes), and any other time that there is a significant change to the contents hereof.

8.0 REFERENCES

Kline, N.W., 1993, "Certification of Version 1.2 of the PORFLO-3 Code for the Hanford CRAY Computer", WHC-SD-ER-CSWD-003 Rev. 0, Westinghouse Hanford Company, Richland WA.

Runchal, A.K., B. Sagar, and N.W. Kline, 1992, "PORFLO-3: A Mathematical Model for Fluid Flow, Heat, and Mass Transport in Variably Saturated Geologic Media. Users Manual, Version 1.2", WHC-EP-0385, Westinghouse Hanford Company, Richland WA.

Appendix A
Description of PRPRW Processor

DON'T SAY IT --- *Write It!*

DATE: September 27, 1993

TO: Niall Kline, Dave Langford

FROM: L. L. Carter

Telephone: 376-3929

SUBJECT: **APPENDIX FOR EDT FOR PORFLO-3 TO QA UPDATE PACKAGE FOR PLATFORMS**

I have written the following rough draft of an appendix for the EDT for PORFLO-3.

PRPR UPDATE FORTRAN PREPROCESSOR FOR COMPUTER PLATFORMS

The PRPR preprocessor code[1] was written by the LANL code developers of MCNP with the goals of simplicity, a reasonable compatibility with update, and FORTRAN that will compile and load on most workstations. This makes it possible to maintain codes with the convenience of update patches on workstations where vendor supplied products are unavailable. Unlike other update emulators, PRPR uses no binary files and is written in portable standard FORTRAN 77. PRPR reads a standard FORTRAN 77 source code from a file named codef, positions common or other code in appropriate locations using (*comdeck and *call) commands, keeps or deletes conditional code with (*define, *if def, *endif) commands, makes modifications using the file named patch, and writes the resulting compile file on a file named compile when *define directives are present in either the codef or patch files.

If the optional patch file is present with more than *define directives, a new codef file called newid is written according to the (*insert, *delete, *before, *ident, *addfile, *deck, and *define) directives in the patch file. The name of this newid file can be switched to codef for a subsequent update.

The principal advantage of PRPR is that it can be used wherever FORTRAN 77 is supported. Other advantages of PRPR are that it is machine-portable, simple (about 200 lines of FORTRAN plus comment cards), and operates directly on the source files codef and patch. The disadvantages and restrictions are:

1. Available commands are limited to those listed in Table 1.
2. All commands in the patch file must be in the same order as the corresponding code in the file codef. For example, changes to deck IM must come before changes to deck HS in patch if IM comes before HS in the file codef.
3. There are very few error traps. If your patch or codef files are wrong, prpr will fail without warning. The few error messages provided are printed at the end of the newid file so you should always look for errors at the end of that file after the execution of prpr.
4. The FORTRAN source file must be named codef and the patch file must be named patch.
5. Files named newid and compile must not be present when prpr is executed.
6. The number of lines in a comdeck and other dimensions are fixed by parameter statements in PRPR and must be increased if exceeded.
7. *define directives must be the first line(s) of either the patch or codef file.
8. *addfile should be immediately followed by *deck on the next line.
9. Either an *ident or an *addfile/*deck must precede *insert, *delete, *before, *edit commands.
10. Nothing can be added after the last line of the codef file.
11. PRPR is slower than HISTORIAN and UPDATE.

The directives recognized by PRPR are given in Table 1.

Table 1. Directives Recognized by PRPR

Long Directive	Abbreviated Form	Function
*/	*/	comment
*define,c,d	*df,c,d	set condition to "c" and "d", etc.
*ident,a	*id,a	change patch identifier to "a"
*edit,a	*e,a	process only deck "a"
*addfile ,a	*af ,a	add subroutines after deck "a"
*deck,a	*dk,a	define deck identifier "a"
*insert,a.n	*i,a.n	insert lines after line "a.n"
*delete,a.m,a.n	*d,a.m,a.n	delete or replace lines "a.m" through "a.n"
*before,a.n	*b,a.n	insert lines before line "a.n"
*comdeck,a	*cd,a	define common deck identifier "a"
*call,a	*ca,a	insert comdeck "a"
*if def,c,n		keep following n lines if condition c is met
*endif	*ei	end conditional if

All commas in the above can be replaced with blanks except for the *addfile where a blank must proceed the comma.

The following are the Rules of Operation of PRPR:

1. If the patch file does not exist, a compile file is produced from codef.
2. If the patch file exists and contains more than *define directives, both newid and compile files will be generated.
3. If no *define directives are present, only newid is produced from codef.
4. The absence of a codef file at execution of prpr means that this is the first creation of a newid with update identifiers. Then the source file, containing *deck "name" or *comdeck "name" cards before each section of code that will be assigned that identifier "name" on each card along with a unique number, is assumed to be contained on the file patch.

Many of the comment cards for PORFLO-3 have comment cards with data extending into columns 73 to 80. A few modifications were made in the PRPR source code, called PRPRW to denote code after these changes, to keep this information in columns 73 to 80. Prior to these changes, PRPR treated columns 73 to 80 as blanks, put the deck (or comdeck or change) identifier in columns 81 to 88, and the numeric card identifier in columns 89 to 93. We have not changed the location of the deck and numeric card identifiers, but the blank columns of columns 73 to 80 are replaced with whatever was on the original source file or corrections on the patch file. The specific changes made in the PRPR FORTRAN file were:

- change "*72" to "*80" in the character statements;
- change the pattern "a72" to "a80" throughout;
- change the pattern "8x," to "," throughout.

The LANL version of PRPR did not allow the user to create his original newid from scratch. We have allowed for this by making the following additional changes in PRPR so that if the file codef does not exist, a first creation of newid is made using the source code on the file patch. The specific additional changes in the PRPR FORTRAN file that were made to allow for this necessary option were:

- change "status='old'" to "status='unknown'" on the "open(3,..." card;
- insert the 10 cards after the "open(5,..." card, where the first 6 of these 10 cards are comment cards. The remaining 4 cards check to see if there is any data on the codef file, and if there isn't any, switch to reading the source code in from the patch file with a set of a parameter ja to 1.

We have had reliable performance from PRPR in MCNP applications here at WHC. It is both simple and easy to use. Applications of PRPRW with PORFLO-3, as discussed in this document [see for example the pp3d"200" patch files and the ucl."platform" files], have reinforced this experience of simplicity and reliability.

REFERENCES

1. Packaged documentation from Radiation Shielding Information Center at ORNL containing the MCNP manual and relevant newsletters from LANL, RSIC COMPUTER CODE COLLECTION -- MCNP4: Monte Carlo Neutron and Photon Transport Code System, CCC-200A/B, June 1991. [see APPENDIX C, pages C-5 and C-6, SECTION 2 of MCNP3B Newsletter for a discussion of prpr.]

Appendix B
Example Update, Compile and Load Script for IBM

```
#! /bin/sh
# Script to update with prpr, compile, and load PORFLO-3 on IBM.
# Input files required from cfs or disk:
#
#     p3d.src    update source file for main code
#     texas.src  update source file for ITPACK routines
#     pp3d       modification (patch) file for main code
#     ptexas     modification (patch) file for ITPACK solvers
#     prprwf     FORTRAN source code for prprw
#     r_timdat.o relocateable for r_timdat.c
#
# File pp3d must be copied from list of existing (sgi) patch files.
# Check the end of newid1, newid2, newid3, and newid for errors
mkdir flib
rm flib/*
mkdir olib
rm olib/*
# compile and load prpr
cp prprwf prprw.f
f77 prprw.f -o prprw
rm prprw.f
# put main code in update format
rm codef compile compile.f newid patch
cp p3d.src patch
prprw
cp newid newid1
mv newid codef
rm compile
# incorporate changes for IBM and produce compile file
rm patch
# change update *def header to ibmx
sed 1,1s/sgix/ibmx/ pp3d >patch
prprw
mv compile compile.f
mv newid newid2
# put texas code in update format
rm codef patch
cp texas.src patch
prprw
cp newid newid3
mv newid codef
rm compile
# produce compile file for texas subroutines
cp ptexas patch
prprw
# concatenate compile file
rm compz.f
cat compile.f compile >compz.f
rm compile.o compile.f compile
```

```
# compile and load
f77 -qsave -O2 -qautodbl=dblpad -o p3d.x *.f r_timdat.o
# move FORTRAN subroutines to sub-directory flib
mv *.f flib
# move FORTRAN object decks to sub-directory olib
mv *.o olib
cp olib/r_timdat.o .
```

Appendix C
Standard Patch File for SGI Cycle Number 200

```
*define sgix
*ident llcsgi
*d PARAM.3,PARAM.4
      PARAMETER ( LX = 100, LY = 4, LZ = 100, LMX = 100 )
      PARAMETER ( LZN=10, LSS=10, LSO=100, LUS=100 )
*d PARAM5.6
      Parameter (MAXSTP = 32000)
*addfile ,CRY1
*comdeck sgi1
      Call IDATE (KTEMP(1),KTEMP(2),KTEMP(3))
         IMONTH=KTEMP(1)
         IDAY=KTEMP(2)
         IYR=MOD(KTEMP(3),100)
      Call ITIME (KTEMP)
         IHOUR=KTEMP(1)
         IMIN=KTEMP(2)
         ISEC=KTEMP(3)
      Write (LDATE,'(I2,1H/,I2,3H/19,I2)') IMONTH,IDAY,IYR
      Write (LTIME,'(I2,1H:,I2,1H:,I2,2H  )') IHOUR,IMIN,ISEC
      NDATE=LDATE
      NTIME=LTIME
*i DATTIM.18
      DIMENSION KTEMP(3)
*i DATTIM.60
c       include EDT QA number on output file
      WRITE (IWR,570)
      WRITE (IWR,501)
*if def,sgix,1
  501 FORMAT (15X,'QA on SGI With EDT Number ')
*if def,sunx,1
  501 FORMAT (15X,'QA on SUN With EDT Number ')
*if def,ibmx,1
  501 FORMAT (15X,'QA on IBM With EDT Number ')
      WRITE (IWR,570)
*d DATTIM.67
*if def,sgix.or.sunx.or.ibmx,1
*call sgi1
*if def,cray,1
*d DATTIM.83
      IRUNID=IMIN+IHOUR*100+IYR*10000+IDAY*1000000+IMONTH*100000000
*d DATTIM.116
*if def,sgix.or.sunx.or.ibmx,1
*call sgi1
*if def,cray,1
*d FLUX1.111
     6 2X,'CONV CUM FLUX',2X,'DIFF CUM FLUX',2X,'TOTAL CUM FLUX')
*d PORFLO.67
      IDSITE   = 200
*if def,sunx,1
      IDSITE=IDSITE+100
*if def,ibmx,1
      IDSITE=IDSITE+200
*i PORFLO.118
*if def,sunx,1
      WRITE (IWR,1101)
*d PORFLO.247
 1000 FORMAT(' ',79('*')
*d PORFLO.271
*if def,sunx,2
     & )
```

```
 1101 FORMAT(13X,' '
*d PRINT.352,PRINT.355
     &  5X,'MAXIMUM NUMBER OF NODES IN EACH 3-D ARRAY. =',I8/
     &  5X,'NUMBER OF THREE DIMENSIONAL FIELD ARRAYS.. =',I8/
     &  5X,'FIELD LENGTH ALLOCATED FOR 3-D ARRAYS..... =',F8.1,' Kbyte'/
     &  5X,'FIELD LENGTH ACTUALLY USED FOR 3-D ARRAYS. =',F8.1,' Kbyte')
*d PRINT.436
     & ,         '  # OF SWEEPS  RELAX FACTOR')
*d ROUND.16
         ALS    = LOG10(FD)
*d TIMSTP.21
            TIMSTP =  LOG(VNEXT / VCRNT) / A1
*d XYPLOT.98
      ALY       = LOG10(YMIN) +1. - YMINDN
*addfile ,ZNAME
*deck se9
      subroutine idate (k1,k2,k3)
c234567      subroutine to return integer dd,mm,yy                l gannon   9/93
c              uses the generic r_date    c routine written by jay painter
c
      integer  k1,k2,k3
      character*8 date
c
      call r_date (date)
      read (date,fmt="(3(i2,1x))") k1,k2,k3
      return
      end
      subroutine itime (kt)
c234567      subroutine to return integer hours, mins, sec       l gannon   9/93
c              uses the generic r_tod  c routine written by jay painter
c
      dimension kt(3)
      integer kt
      character*8 tod
c
      call r_tod (tod)
      read (tod,fmt="(3(i2,1x))") kt(1),kt(2),kt(3)
      return
      end
      subroutine second(t)
c              use the generic richland r_etime64  c routine
      call r_etime64(t)
      return
      end
      subroutine tremain (t)
c234567  use the generic richland r_tremain64  c routine
      call r_tremain64(t)
      return
      end
```