

1/95

SOFTWARE RELEASE NOTICE

| | | |
|--|----------------------|--------------------------|
| 1. SRN Number: PA-SRN-204 | | |
| 2. Project Title: TSPA & Technical Integration Code | | Project No. 20-01402-762 |
| 3. SRN Title: TPA Version 3 <i>10-4-99</i> | | |
| 4. Originator/Requestor: Ron Janetzke | | Date: Oct. 1, 1999 |
| 5. Summary of Actions <ul style="list-style-type: none"> <input type="checkbox"/> Release of new software <input type="checkbox"/> Release of modified software: <ul style="list-style-type: none"> <input type="checkbox"/> Enhancements made <input type="checkbox"/> Corrections made <input type="checkbox"/> Change of access software <input checked="" type="checkbox"/> Software Retirement | | |
| 6. Persons Authorized Access | | |
| Name | Read Only/Read-Write | Addition/Change/Delete |
| Ron Janetzke | RW | |
| Sitakanta Mohanty | RW | |
| Tim McCartin (NRC) | RW | |
| M. Rose Byrne (NRC) | RW | |
| 7. Element Manager Approval: <i> Gordon Wilkerson </i> | | Date: 10/4/99 |
| 8. Remarks: Version superceded. | | |

3/95

SOFTWARE SUMMARY FORM

| | | | |
|--|---|---|--|
| 01. Summary Date: 04/16/97 | 02. Summary prepared by (Name and phone) B. Mabrito (210) 522-5149 | 03. Summary Action: New | |
| 04. Software Date: 03/17/97 | 05. Short Title: TPA Version 3.0 | | |
| 06. Software Title: TPA - Total-System Performance Assessment Computer Code, Version 3.0 | | 07. Internal Software ID: None | |
| 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 type="checkbox"/> Scientific/Engineering <input type="checkbox"/> Auxiliary Analyses <input checked="" type="checkbox"/> Total System PA <input type="checkbox"/> Subsystem PA <input type="checkbox"/> Other b. Specific: | |
| 11. Submitting Organization and Address: CNWRA/SwRI 6220 Culebra Road San Antonio, TX 78228 | | 12. Technical Contact(s) and Phone: Sitakanta Mohanty (210) 522-5185 | |
| 13. Narrative: The TPA Code consists of the following modules: UZFLOW, NFENV, EBSREL, UZFT, SZFT, DCAGW, FAULTO, SEISMO, VOLCANO, ASHPLUMO, ASHRMVO, DCAGS, LHS EXEC. | | | |
| 14. Computer Platform SUN Workstation | 15. Computer Operating System: UNIX | 16. Programming Language(s): FORTRAN | 17. Number of Source Program Statements: approx. 30,000 LINES |
| 18. Computer Memory Requirements: Unknown | 19. Tape Drives: None | 20. Disk/Drum Units: N/A | 21. Graphics: Unknown |
| 22. Other Operational Requirements None | | | |
| 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 | |
| Software Custodian: <u>B. Mabrito</u> Date: <u>4/16/97</u> | | | |

4/95

SOFTWARE RELEASE NOTICE

| | | |
|--|--|----------------------------------|
| 01. SRN Number: PA-SRN-134 | | |
| 02. Project Title: TSPA & Technical Integration Code | | Project No. 20-5708-762 |
| 03. SRN Title: TPA Version 3.0 | | |
| 04. Originator/Requestor: Bruce Mabrito | | Date: 04/16/97 |
| 05. Summary of Actions <input checked="" 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 type="checkbox"/> Software Retirement | | |
| 06. Persons Authorized Access | | |
| Name | RO/RW | A/C/D |
| Sitakanta Mohanty Ron Janetzke Dennis Vinson M. Rose Byrne | READ/WRITE READ/WRITE READ/WRITE READ/WRITE | ADDED ADDED ADDED ADDED |
| 07. Element Manager Approval: <i>[Signature]</i> | | Date: 4/16/97 |
| 08. Remarks: TPA Version 3.0 software was hand-delivered and installed at NRC Headquarters 3/17/97 by a CNWRA staff member. | | |

6/95

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE: TPA VERSION 3.0

April 14, 1997

TOTAL PERFORMANCE ASSESSMENT VERSION 3.0

1. Scientific Notebook Documentation Development: CNWRA Electronic Scientific Notebooks numbered: 170, S. Mohanty and R. Janetzke - EBSFAIL, EBSREL, UZFT, and SZFT; 164, M. Jarzemba - ASHPLUMO; 202, R. Manteufel and S. Mohanty - EXEC, SEISMO, DCAGS, DCAGW, VOLCANO, NFENV, and UTILITY modules; 167, A. Ghosh - FAULTO; 196, S. Hsiung, SEISMO; 163, S. Stothoff - UZFLOW and CLIMATO; 115, C. Connor and B. Hill - VOLCANO; 190, R. Janetzke, TPA v. 2.0 and Audit 96. These notebooks document the software development. All scientific notebooks have been reviewed by the cognizant EMs and were verified by the software custodian.
2. Programming Language: ANSI Standard FORTRAN 77 confirmed by the software custodian.
3. Internal Documentation: Comments in the subroutine of VOLCANO were observed by the software custodian and R. Rice on 4/9/97. The TPA Version 3.0 scientific & engineering software was on the Bigbend server. The code contained comments and a hard copy showing representative comments is included in attached documentation. Additionally, comments were checked in the TPA Version 3.0 EXEC main program and were found to be clear and numerous. The TPA Version 3.0 code is designed so that internal code documentation can be easily extracted using a supporting program. The internal documentation comments in TPA Version 3.0 meet the requirements of TOP-018, Section 5.4.4.
4. Software Labels and Data
 - a. Header Data and Format: TPA Version 3.0 has header information which does not fully meet the TOP-018 requirements. The TPA Version 3.0 software was delivered to the NRC 3/17/97 with the understanding that the NRC and the CNWRA would exercise the software and propose changes to the code which will result in the TPA Version 3.1 software. That version of TPA will be sent to the NRC on April 30, 1997. R. Rice prepared a header which contains the required information and format style for the TPA Version 3.1 code. It complies with the TOP-018 requirements of Section 5.4.6 and will be reflected in TPA Version 3.1.
 - b. NRC Data: The TPA Version 3.0 NRC data header section was not in full compliance with TOP-018. However, it will be updated in TPA Version 3.1 and will be consistent with TOP-018, Section 5.4.6, fourth bullet.
 - c. Source Code Header: The source code header data was compared to TOP-018, Section 5.4.6, fifth bullet and was found to not completely meet requirements. Header data meeting the TOP-018 requirements will be included in TPA Version 3.1.

7/95

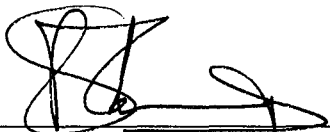
5. Unique Run Identification. At the top of each output file, a unique identifier states information such as the following: "TPA 3.0, Job started: Wed Apr 9 10:49:40 1997". This identifier meets the requirements of unique run identification.

6. Software Analysis and Results

a. Analysis: The software analysis tool FOR_STUDY Version 1.2 was utilized on TPA Version 3.0, but only after it was received by the CNWRA in late March, 1997. It was not available for use on TPA Version 3.0 before the code was delivered to the NRC. FOR_STUDY was run in the default configuration as specified in its User's Manual, with the exception that the FORTRAN compiler was specified as "Sun." The software analysis tool was run on TPA Version 3.0 on 4/8/97 by P. Starkweather. Software analysis tools/compiler such as the static analyzer SPAG contained in plus_FORT, and MICROSOFT and LAHEY FORTRAN compilers were run with TPA Version 3.0 source code throughout its development.

b. Analysis Report: FOR_STUDY reported a total of 1,506 warning and error messages were noted on TPA 3.0 and 28 lines were noted as having warning and error messages on generating the interface. For subcomponents of the TPA 3.0 code, FOR_STUDY listed the following warning and error messages: 12 lines in the ASHPLUME interface and 26 in ASHPLUME; 5 lines in the FAILT interface and 21 in FAILT; 6 lines in the NEFMKS interface and 137 in NEFMKS; 5 lines in the RELEASET interface and 37 in RELEASET; 2 lines in the SNLLHS interface and 95 in SNLLHS. FOR_STUDY provided a total of 346 warning and error messages in the TPA 3.0 subcomponent files, including duplicates.

c. Resolution of Comments: Since the FOR_STUDY analysis tool program was not available for the 3/17/97 delivery, the earlier use of other software analysis tools was helpful in debugging the TPA Version 3.0 code. The warning and error messages from those studies were evaluated and necessary modifications were made to the TPA Version 3.0 source code. Results from later FOR_STUDY runs were provided to the cognizant developer for resolution in the TPA Version 3.1 code. Many of the returned error/warning messages were due to the lack of an interface file for the Sun Fortran compiler. Most errors are viewed as nonfatal and these errors are not recognized by the SUN compiler.



CNWRA Software Developer



CNWRA Software Custodian

Attachments/

original to: Software Folder
cc: CNWRA Software Developer

8/95

```
OBJECTS = array.o \
ashplumo.o \
ashrmovo.o \
dcagw.o \
dcags.o \
ebsfail.o \
ebsrel.o \
fault.o \
invent.o \
mv.o \
nfenv.o \
szft.o \
reader.o \
sampler.o \
seismo.o \
subarea.o \
uzflow.o \
uzft.o \
volcano.o \
ran.o \
numrecip.o \
condxyzt.o \
peakfinder.o \
fileunit.o \
findelev.o
```

TPA Version 3.0
make file

```
szft.o : szft.f
```

```
    f77 -c -N1200 szft.f
```

```
tpa : $(OBJECTS)
```

```
    f77 -g exec.f -o tpa.e $(OBJECTS)
```

9/95

```

c=====
      program exec
c=====
c Executive for TPA Version 3.0
c by Randall D. Manteufel, January 27, 1997
c
cc      1      2      3      4      5      6      7
cc34567890123456789012345678901234567890123456789012345678901234567890
c
      implicit double precision (a-h,o-z)
      implicit integer (i-n)

      include 'maxntime.i'
      include 'maxnsuba.i'
      include 'maxnnucl.i'
      parameter (maxseismicevents = 150 )

      character*6 names(maxnnucl)
      character*60 name
      character*160 mes1
      character*160 mes2

      dimension tim(maxntime)
      dimension qm3peryrinsa( maxntime )
      dimension qm3peryrperwpinsa( maxntime )
      dimension temprep( maxntime )
      dimension tempwp( maxntime )
      dimension tempsf( maxntime )
      dimension relhumwp( maxntime )
      dimension phwp( maxntime )
      dimension clwp( maxntime )
      dimension qm3peryrperwpinsamisswp( maxntime )
      dimension qm3peryrperwpinsahitwp( maxntime )
      dimension corrosionfailwp( maxntime )
      dimension seismicfailwp( maxntime )
      dimension faultfailwp( maxntime, maxnsubarea )
      dimension volcanicfailwp( maxntime, maxnsubarea )
      dimension volcanicamtufailed( maxnsubarea )
      dimension ciperyrperwpinsafromwp( maxntime, maxnnucl )
cc      dimension qm3peryrinsaintoloweruz( maxntime )
      dimension ciperyrinsaintoloweruz( maxntime, maxnnucl )
      dimension qm3peryrinsafromuz( maxntime )
      dimension ciperyrinsafromuz( maxntime, maxnnucl )
cc-new
      dimension qm3peryrinsafromsz( maxntime )
      dimension ciperyrinsafromsz( maxntime, maxnnucl )
      dimension qm3peryrallsafromsz( maxntime )
      dimension ciperyrallsafromsz( maxntime, maxnnucl )

```

10/95

```

=====
      subroutine volcano( nsa, mxntim, ntim, tim,
      & amtufail, amtuejected, pfail )
=====
c determine direct disruptive consequence of volcanism at YM due to both
c   1) intrusive event that fails waste packages
c   2) extrusive event that ejects waste in an ash plume
c by Randall D. Manteufel, January 27, 1997

c nsa          = input, integer, number of subarea for repository
c mxntim       = input, integer, maximum number of time steps
c              this is used to dimension arrays
c ntim         = input, integer, number of time steps
c tim(ntim)    = input, double precision, array of times
c amtufail(nsa) = output, double precision, number of WP fail
c              in each subarea due to intrusive event
c amtuejected  = output, double precision, MTU of waste ejected
c              in extrusive event from all subareas
c pfail(mxntim,nsa) = output, double precision, array of percent WP
c              failures due to volcanism in each subarea
c
      implicit double precision (a-h,o-z)

      include 'maxntime.i'
      include 'maxnnucl.i'
      include 'maxnsuba.i'

      dimension tim(ntim)
      dimension amtufail(nsa)
      dimension pfail(mxntim,nsa)

      dimension xyp1(2)
      dimension xyp2(2)
      dimension xyp(2)
      dimension t(4)
      dimension y(4)

      character*60 name

      save ikey
      save aml
      save ixloc
      save iyloc
      save iextint
      save ifractext
      save idiam
      save ilength
      save iangle
      save iwidth
      save ivolctim

      external ivaluesp
      external valuesp

```

11/95

Example input file, supplied with TPA Version 3.1 Code

Supplied to NRC on April 30, 1997

TPA 3.0, Job started: Wed Apr 9 10:49:40 1997

Summed Normalized Release over 10,000 years

Realization EPA_Release_Limit

| | |
|---|-------------|
| 1 | 0.10903E+00 |
| 2 | 0.11751E+00 |
| 3 | 0.12126E+00 |

12/95

Example input file, supplied with TPA Version 3.1 Code
Supplied to NRC on April 30, 1997
TPA 3.0, Job started: Wed Apr 9 10:49:40 1997
AEDE[rem/yr], GroundSurface Pathway
summed over all nuclides, averaged over all realizations

| | |
|-----------|-----------|
| 0.231E+01 | 0.100E-14 |
| 0.467E+01 | 0.100E-14 |
| 0.709E+01 | 0.100E-14 |
| 0.957E+01 | 0.100E-14 |
| 0.121E+02 | 0.100E-14 |
| 0.147E+02 | 0.100E-14 |
| 0.174E+02 | 0.100E-14 |
| 0.201E+02 | 0.100E-14 |
| 0.228E+02 | 0.100E-14 |
| 0.257E+02 | 0.100E-14 |
| 0.286E+02 | 0.100E-14 |
| 0.316E+02 | 0.100E-14 |
| 0.346E+02 | 0.100E-14 |
| 0.378E+02 | 0.100E-14 |
| 0.409E+02 | 0.100E-14 |
| 0.442E+02 | 0.100E-14 |
| 0.476E+02 | 0.100E-14 |
| 0.510E+02 | 0.100E-14 |
| 0.545E+02 | 0.100E-14 |
| 0.581E+02 | 0.100E-14 |
| 0.617E+02 | 0.100E-14 |
| 0.655E+02 | 0.100E-14 |
| 0.693E+02 | 0.100E-14 |
| 0.733E+02 | 0.100E-14 |
| 0.773E+02 | 0.100E-14 |
| 0.814E+02 | 0.100E-14 |
| 0.856E+02 | 0.100E-14 |
| 0.900E+02 | 0.100E-14 |
| 0.944E+02 | 0.100E-14 |
| 0.989E+02 | 0.100E-14 |
| 0.104E+03 | 0.100E-14 |
| 0.108E+03 | 0.100E-14 |
| 0.113E+03 | 0.100E-14 |
| 0.118E+03 | 0.100E-14 |
| 0.123E+03 | 0.100E-14 |
| 0.128E+03 | 0.100E-14 |
| 0.134E+03 | 0.100E-14 |
| 0.139E+03 | 0.100E-14 |
| 0.145E+03 | 0.100E-14 |
| 0.150E+03 | 0.100E-14 |
| 0.156E+03 | 0.100E-14 |
| 0.162E+03 | 0.100E-14 |
| 0.168E+03 | 0.100E-14 |
| 0.174E+03 | 0.100E-14 |
| 0.181E+03 | 0.100E-14 |
| 0.187E+03 | 0.100E-14 |
| 0.194E+03 | 0.100E-14 |
| 0.201E+03 | 0.100E-14 |
| 0.208E+03 | 0.100E-14 |
| 0.215E+03 | 0.100E-14 |
| 0.223E+03 | 0.100E-14 |
| 0.230E+03 | 0.100E-14 |
| 0.238E+03 | 0.100E-14 |
| 0.246E+03 | 0.100E-14 |
| 0.254E+03 | 0.100E-14 |

pcl5

JOB 354

bruce

For: bigbend!mohanty
Date: Wed Apr 9 15:25:16 CDT 1997
Submit queue: IF 1 / Ethernet / UHSW
Submitted: Fri Feb 14 07:16:50 1992
Started: Fri Feb 14 07:16:50 1992



QMS 3825 Print System

QMS 3825 Print System

1506 lines on tpa 3.0 project
28 lines on generating the interface

14/95

On the subcomponents:

12 ashplume interface
26 ashplume
5 fault interface
21 fault
6 nefmks interface
137 nefmks
5 releaset interface
37 releaset
2 snllhs interface
95 snllhs
346 total, which include duplicates

TPA 3.0

FOR-STUDY RESULTS

15/95

Warning #701 ashplume.f,396: function DINNERA() return type differs with interface
Warning #686 ashplume.f,396: different type and size used in arg #1 of QROMB1()
Warning #701 ashplume.f,421: function DINNERF() return type differs with interface
Warning #701 ashplume.f,442: function DINTEGRANDF() return type differs with interface
Warning #686 ashplume.f,442: different type and size used in arg #1 of QGAUS()
Warning #701 ashplume.f,462: function DINTEGRANDA() return type differs with interface
Warning #686 ashplume.f,462: different type and size used in arg #1 of QGAUS()
Warning #701 ashplume.f,725: function FFINTEGRAND() return type differs with interface
Warning #701 ashplume.f,1120: function FFINTEGRANDHIST() return type differs with interface
Warning #686 ashplume.f,1312: different type and size used in arg #1 of TRAPZD4()
Warning #686 ashplume.f,1416: different type and size used in arg #1 of TRAPZD3()
Warning #686 ashplume.f,1518: different type and size used in arg #1 of TRAPZD2()

Samples of
The FOR-STudy
print outs.

GS

4/16/97

Error #132 ashplume.f,161: DO index, K, can't be changed inside DO loop
Error #132 ashplume.f,168: DO index, K, can't be changed inside DO loop
Warning #72 ashplume.f,203: common /ONE/ was not used in function ASHPLOME()
Warning #72 ashplume.f,203: common /TWO/ was not used in function ASHPLOME()
Warning #72 ashplume.f,203: common /SEVEN/ was not used in function ASHPLOME()
Warning #72 ashplume.f,203: common /EIGHT/ was not used in function ASHPLOME()
Warning #72 ashplume.f,203: common /ELEVEN/ was not used in function ASHPLOME()
Warning #72 ashplume.f,203: common /TWELVE/ was not used in function ASHPLOME()
Warning #72 ashplume.f,308: common /FOUR/ was not used in function OUTHEADER()
Warning #72 ashplume.f,308: common /SIX/ was not used in function OUTHEADER()
Warning #72 ashplume.f,375: common /FOUR/ was not used in function USERINPUT()
Warning #72 ashplume.f,375: common /SIX/ was not used in function USERINPUT()
Warning #701 ashplume.f,396: function DINNERA() return type differs with interface
Warning #686 ashplume.f,396: different type and size used in arg #1 of QROMB1()
Warning #701 ashplume.f,421: function DINNERF() return type differs with interface
Warning #72 ashplume.f,425: common /THIRTEEN/ was not used in function ASHCALCF()
Warning #701 ashplume.f,442: function DINTEGRANDF() return type differs with interface
Warning #686 ashplume.f,442: different type and size used in arg #1 of QGAUS()
Warning #701 ashplume.f,462: function DINTEGRANDA() return type differs with interface
Warning #686 ashplume.f,462: different type and size used in arg #1 of QGAUS()
Warning #701 ashplume.f,725: function FFINTEGRAND() return type differs with interface
Warning #72 ashplume.f,739: common /SEVEN/ was not used in function FF()
Warning #701 ashplume.f,1120: function FFINTEGRANDHIST() return type differs with interface
Warning #686 ashplume.f,1312: different type and size used in arg #1 of TRAPZD4()
Warning #686 ashplume.f,1416: different type and size used in arg #1 of TRAPZD3()
Warning #686 ashplume.f,1518: different type and size used in arg #1 of TRAPZD2()

17/95

ERROR fault.f,322: expecting expr/obj in the READ/WRITE i/o list
ERROR fault.f,433: expecting expr/obj in the READ/WRITE i/o list
ERROR fault.f,463: expecting expr/obj in the READ/WRITE i/o list
ERROR fault.f,464: expecting expr/obj in the READ/WRITE i/o list
Error #670 fault.f,1047: intrinsic MAX() args differ - REAL*4 vs. *8

18/95

Warning #700 fault.f,161: No interface for routine FDATE()
Warning #713 fault.f,268: expecting lvalue as scalar argument - constant/expr passed in arg #
Warning #701 fault.f,269: function RATE() return type differs with interface
Warning #686 fault.f,269: different type and size used in arg #12 of ODEINT()
Warning #701 fault.f,269: function RKQC() return type differs with interface
Warning #686 fault.f,269: different type and size used in arg #13 of ODEINT()
ERROR fault.f,322: expecting expr/obj in the READ/WRITE i/o list
ERROR fault.f,433: expecting expr/obj in the READ/WRITE i/o list
ERROR fault.f,463: expecting expr/obj in the READ/WRITE i/o list
ERROR fault.f,464: expecting expr/obj in the READ/WRITE i/o list
Warning #434 fault.f,588: parameter ZERO definition & declared type differ
Warning #434 fault.f,714: parameter FCOR definition & declared type differ
Warning #434 fault.f,714: parameter ONE definition & declared type differ
Warning #434 fault.f,714: parameter SAFETY definition & declared type differ
Warning #434 fault.f,714: parameter ERRCON definition & declared type differ
Warning #72 fault.f,838: common /WETTING/ was not used in function RK4()
Warning #434 fault.f,852: parameter ONE definition & declared type differ
Warning #434 fault.f,852: parameter SHRINK definition & declared type differ
Warning #434 fault.f,852: parameter GROW definition & declared type differ
Warning #686 fault.f,874: different type and size used in arg #8 of MMID()
Error #670 fault.f,1047: intrinsic MAX() args differ - REAL*4 vs. *8

19/95

Warning #103 nefmks.f,1205: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,1205: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,1206: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,1206: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,1206: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,1206: expecting REAL constant in DATA - not REAL*8

Warning #700 nefmks.f,522: no interface for routine FDATE()
Warning #543 nefmks.f,746: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,758: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,850: loss of precision in assignment: REAL*8 to REAL*4
Warning #103 nefmks.f,1205: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,1205: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,1206: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,1206: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,1206: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,1206: expecting REAL constant in DATA - not REAL*8
Warning #543 nefmks.f,1352: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,1364: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #72 nefmks.f,1558: common /OUTVC/ was not used in function SETUP()
Warning #72 nefmks.f,1558: common /LEGS/ was not used in function SETUP()
Warning #72 nefmks.f,1558: common /BFIX/ was not used in function SETUP()
Warning #72 nefmks.f,1558: common /XPORTA/ was not used in function SETUP()
Warning #543 nefmks.f,1770: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #72 nefmks.f,2014: common /XPORTA/ was not used in function FACER()
Warning #543 nefmks.f,2083: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2092: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2338: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2339: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2356: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2358: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2414: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2415: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2433: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2436: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2501: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2504: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2514: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2535: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2539: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2548: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2582: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2586: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2626: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2634: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2635: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2641: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2750: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2751: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2754: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,2774: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,2846: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,2847: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,3118: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3129: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3137: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3151: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3171: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3174: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3196: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3292: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3458: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3810: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3832: loss of precision in assignment: REAL*8 to REAL*4
Warning #72 nefmks.f,4058: common /XPORTB/ was not used in function PTHLEN()
Warning #543 nefmks.f,4266: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,4280: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,4284: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,4285: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,4335: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,4341: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,4348: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,4357: loss of precision in assignment: REAL*8 to REAL*4

Warning #543 nefmks.f,4364: loss of precision in assignment: REAL*8 to REAL*4
Warning #72 nefmks.f,5408: common /XPORTB/ was not used in function STOPPER()
Warning #543 nefmks.f,5991: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,6007: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,6042: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,6060: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,6061: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,6073: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,6076: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,6103: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,6110: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #72 nefmks.f,6127: common /SOLIMT/ was not used in function CHAIN()
Warning #72 nefmks.f,6127: common /XPORTA/ was not used in function CHAIN()
Warning #543 nefmks.f,6263: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6276: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6283: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6448: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6467: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6757: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,7072: loss of precision in assignment: REAL*4 to INTEGER*4
Error #171 nefmks.f,7401: Local name LTPA was referenced but never set
Warning #72 nefmks.f,7401: common /DENSER/ was not used in function GETRV()
Warning #72 nefmks.f,7401: common /FLOW/ was not used in function GETRV()
Warning #72 nefmks.f,7401: common /ISOTOP/ was not used in function GETRV()
Warning #72 nefmks.f,7401: common /JUNCT/ was not used in function GETRV()
Warning #72 nefmks.f,7401: common /MISC/ was not used in function GETRV()
Warning #72 nefmks.f,7401: common /OPTION/ was not used in function GETRV()
Warning #72 nefmks.f,7401: common /NEFFIL/ was not used in function GETRV()
Warning #72 nefmks.f,7401: common /INAME/ was not used in function GETRV()
Warning #72 nefmks.f,7401: common /LEGS/ was not used in function GETRV()
Warning #72 nefmks.f,7401: common /QSOURCE/ was not used in function GETRV()
Warning #72 nefmks.f,7401: common /RETARD/ was not used in function GETRV()
Warning #72 nefmks.f,7401: common /SOLIMT/ was not used in function GETRV()
Warning #72 nefmks.f,7401: common /VELFLD/ was not used in function GETRV()
Warning #72 nefmks.f,7401: common /XFER/ was not used in function GETRV()
Warning #72 nefmks.f,7587: common /DENSER/ was not used in function RDSMPL()
Warning #72 nefmks.f,7587: common /FLOW/ was not used in function RDSMPL()
Warning #72 nefmks.f,7587: common /ISOTOP/ was not used in function RDSMPL()
Warning #72 nefmks.f,7587: common /JUNCT/ was not used in function RDSMPL()
Warning #72 nefmks.f,7587: common /OPTION/ was not used in function RDSMPL()
Warning #72 nefmks.f,7587: common /OUTVC/ was not used in function RDSMPL()
Warning #72 nefmks.f,7587: common /INAME/ was not used in function RDSMPL()
Warning #72 nefmks.f,7587: common /LEGS/ was not used in function RDSMPL()
Warning #72 nefmks.f,7587: common /QSOURCE/ was not used in function RDSMPL()
Warning #72 nefmks.f,7587: common /RETARD/ was not used in function RDSMPL()
Warning #72 nefmks.f,7587: common /VELFLD/ was not used in function RDSMPL()
Warning #72 nefmks.f,7587: common /XFER/ was not used in function RDSMPL()
Warning #72 nefmks.f,7587: common /XPORTB/ was not used in function RDSMPL()
Warning #543 nefmks.f,7934: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 nefmks.f,7935: loss of precision in assignment: REAL*8 to REAL*4
Warning #72 nefmks.f,8324: common /INAME/ was not used in function METHOD()
Warning #543 nefmks.f,8389: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,8396: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,8399: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,8404: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,8408: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,8409: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,8411: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,8459: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,8465: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,8472: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,8474: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,8475: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,8476: loss of precision in assignment: REAL*8 to REAL*4
Warning #72 nefmks.f,9108: common /MISC/ was not used in function SRCIN()

Warning #715 nefmks.f,9304: expecting 2-dimension array - 1-dimension array passed in arg #4
Warning #715 nefmks.f,9382: expecting 2-dimension array - 1-dimension array passed in arg #4
Warning #718 nefmks.f,9382: expecting array - variable passed in arg #5 of TIMVAL()
Warning #72 nefmks.f,9643: common /SOLIMT/ was not used in function NEFDIT()
Warning #72 nefmks.f,9725: common /SOLIMT/ was not used in function WRTDR2()

ERROR releaset.f,530: expecting expr/obj in the READ/WRITE i/o list
ERROR releaset.f,1176: expecting expr/obj in the READ/WRITE i/o list
ERROR releaset.f,1185: expecting expr/obj in the READ/WRITE i/o list
ERROR releaset.f,1186: expecting expr/obj in the READ/WRITE i/o list
ERROR releaset.f,2821: expecting expr/obj in the READ/WRITE i/o list

Warning #700 releaset.f,339: No interface for routine FDATE()
ERROR releaset.f,530: expecting expr/obj in the READ/WRITE i/o list
Warning #682 releaset.f,787: different size reals used in arg #1 of DECAY()
Warning #682 releaset.f,849: different size reals used in arg #1 of DECAY()
Warning #72 releaset.f,919: common /ICARBON/ was not used in function RELEASE()
Warning #72 releaset.f,919: common /RKUTTA/ was not used in function RELEASE()
Warning #72 releaset.f,919: common /CUMU/ was not used in function RELEASE()
ERROR releaset.f,1176: expecting expr/obj in the READ/WRITE i/o list
ERROR releaset.f,1185: expecting expr/obj in the READ/WRITE i/o list
ERROR releaset.f,1186: expecting expr/obj in the READ/WRITE i/o list
Warning #701 releaset.f,1313: function DERIVS() return type differs with interface
Warning #686 releaset.f,1313: different type and size used in arg #12 of ODEINT()
Warning #701 releaset.f,1313: function RKQC() return type differs with interface
Warning #686 releaset.f,1313: different type and size used in arg #13 of ODEINT()
Error #132 releaset.f,1315: DO index, IT, can't be changed inside DO loop
Warning #72 releaset.f,1835: common /DIFREL1/ was not used in function DERIVS()
Warning #72 releaset.f,1835: common /ARRAY1/ was not used in function DERIVS()
Warning #72 releaset.f,2066: common /CNUCL3/ was not used in function DFLUX()
Warning #72 releaset.f,2066: common /ARRAY1/ was not used in function DFLUX()
Warning #72 releaset.f,2266: common /UO24WIDTH/ was not used in function GASREL()
Warning #72 releaset.f,2454: common /CWAST1/ was not used in function ODEINT()
Warning #434 releaset.f,2478: parameter FCOR definition & declared type differ
Warning #434 releaset.f,2478: parameter ONE definition & declared type differ
Warning #434 releaset.f,2478: parameter ERRCON definition & declared type differ
Warning #72 releaset.f,2592: common /CWAST1/ was not used in function RKQC()
Warning #72 releaset.f,2673: common /CWAST1/ was not used in function RK4()
ERROR releaset.f,2821: expecting expr/obj in the READ/WRITE i/o list
Warning #70 releaset.f,2840: common /UO24WIDTH/ changed relative to prior use
Warning #71 releaset.f,2840: common /UO24WIDTH/ changed relative to first usage
Warning #72 releaset.f,3309: common /CNUCL3/ was not used in function SETDIF()
Warning #72 releaset.f,3309: common /ARRAY1/ was not used in function SETDIF()
Warning #72 releaset.f,3439: common /CCONT2/ was not used in function INIT()
Error #132 releaset.f,3674: DO index, IT, can't be changed inside DO loop
Warning #72 releaset.f,3860: common /CNUCL3/ was not used in function INPUT()
Warning #72 releaset.f,3860: common /ARRAY1/ was not used in function INPUT()
Warning #72 releaset.f,4012: common /DIFREL1/ was not used in function RDELEM()
Warning #72 releaset.f,4012: common /ARRAY1/ was not used in function RDELEM()

25/95

Warning #170 snllhs.f,1249: function name, GAMALN, was not set

Warning #103 snllhs.f,4943: expecting REAL constant in DATA - not CHARACTER

26/95

Warning #700 snllhs.f,82: No interface for routine IEEE_HANDLER()
Warning #700 snllhs.f,369: No interface for routine EXIT()
Warning #72 snllhs.f,394: common /WORKX/ was not used in function LHS()
Warning #72 snllhs.f,457: common /SAMP/ was not used in function BANNER()
Warning #713 snllhs.f,491: expecting lvalue as scalar argument - constant/expr passed in arg
Warning #713 snllhs.f,491: expecting lvalue as scalar argument - constant/expr passed in arg
Warning #701 snllhs.f,501: function BETAFN() return type differs with interface
Warning #543 snllhs.f,653: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 snllhs.f,752: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #626 snllhs.f,798: differing number of GAMALN() arguments; intrf. vs. call
Warning #543 snllhs.f,817: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #543 snllhs.f,822: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #719 snllhs.f,1030: expecting array (to set) - string constant passed in arg #2 of ER
Warning #684 snllhs.f,1030: mixed strings and arithmetic objects in arg #2 of ERRCHK()
Warning #719 snllhs.f,1032: expecting array (to set) - string constant passed in arg #2 of ER
Warning #684 snllhs.f,1032: mixed strings and arithmetic objects in arg #2 of ERRCHK()
Warning #719 snllhs.f,1034: expecting array (to set) - string constant passed in arg #2 of ER
Warning #684 snllhs.f,1034: mixed strings and arithmetic objects in arg #2 of ERRCHK()
Warning #719 snllhs.f,1036: expecting array (to set) - string constant passed in arg #2 of ER
Warning #684 snllhs.f,1036: mixed strings and arithmetic objects in arg #2 of ERRCHK()
Error #714 snllhs.f,1043: expecting array argument - not scalar expression in arg #2 of ERRC
Warning #684 snllhs.f,1043: mixed strings and arithmetic objects in arg #2 of ERRCHK()
Warning #543 snllhs.f,1183: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #719 snllhs.f,1236: expecting array (to set) - string constant passed in arg #2 of ER
Warning #684 snllhs.f,1236: mixed strings and arithmetic objects in arg #2 of ERRCHK()
Error #714 snllhs.f,1244: expecting array argument - not scalar expression in arg #2 of ERRC
Warning #684 snllhs.f,1244: mixed strings and arithmetic objects in arg #2 of ERRCHK()
Warning #170 snllhs.f,1249: function name, GAMALN, was not set
Warning #72 snllhs.f,1355: common /SAMP/ was not used in function CHKDIM()
Warning #72 snllhs.f,1435: common /PARAM/ was not used in function CHKSTR()
Warning #72 snllhs.f,1435: common /SAMP/ was not used in function CHKSTR()
Warning #72 snllhs.f,1543: common /SAMP/ was not used in function CHLSKY()
Warning #543 snllhs.f,1599: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #72 snllhs.f,1641: common /SAMP/ was not used in function CMCRD()
Warning #72 snllhs.f,1690: common /SAMP/ was not used in function CONSTA()
Error #714 snllhs.f,2500: expecting array argument - not scalar expression in arg #2 of ERRC
Warning #684 snllhs.f,2500: mixed strings and arithmetic objects in arg #2 of ERRCHK()
Warning #719 snllhs.f,2502: expecting array (to set) - string constant passed in arg #2 of ER
Warning #684 snllhs.f,2502: mixed strings and arithmetic objects in arg #2 of ERRCHK()
Warning #719 snllhs.f,2504: expecting array (to set) - string constant passed in arg #2 of ER
Warning #684 snllhs.f,2504: mixed strings and arithmetic objects in arg #2 of ERRCHK()
Warning #719 snllhs.f,2506: expecting array (to set) - string constant passed in arg #2 of ER
Warning #684 snllhs.f,2506: mixed strings and arithmetic objects in arg #2 of ERRCHK()
Warning #72 snllhs.f,2649: common /SAMP/ was not used in function HISTO()
Warning #72 snllhs.f,2715: common /SAMP/ was not used in function HPSRT()
Warning #543 snllhs.f,2786: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #719 snllhs.f,2798: expecting array (to set) - string constant passed in arg #2 of ER
Warning #684 snllhs.f,2798: mixed strings and arithmetic objects in arg #2 of ERRCHK()
Error #171 snllhs.f,3236: Local name OUTPUT was referenced but never set
Warning #72 snllhs.f,3522: common /SAMP/ was not used in function MATINV()
Warning #543 snllhs.f,3589: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #626 snllhs.f,3638: differing number of MATINV() arguments; intrf. vs. call
Warning #684 snllhs.f,3784: mixed strings and arithmetic objects in arg #2 of ERRPRT()
Warning #72 snllhs.f,3855: common /SAMP/ was not used in function OUTCRD()
Warning #72 snllhs.f,4116: common /SAMP/ was not used in function PMTRX()
Warning #72 snllhs.f,4185: common /PARAM/ was not used in function POSDEF()
Warning #72 snllhs.f,4185: common /SAMP/ was not used in function POSDEF()
Warning #719 snllhs.f,4407: expecting array (to set) - string constant passed in arg #1 of XE
Warning #684 snllhs.f,4407: mixed strings and arithmetic objects in arg #1 of XERROR()
Warning #713 snllhs.f,4407: expecting lvalue as scalar argument - constant/expr passed in arg
Warning #713 snllhs.f,4407: expecting lvalue as scalar argument - constant/expr passed in arg
Warning #713 snllhs.f,4407: expecting lvalue as scalar argument - constant/expr passed in arg
Warning #72 snllhs.f,4502: common /SAMP/ was not used in function RANKER()
Warning #436 snllhs.f,4573: parameter PC char constant definition will be truncated
Error #171 snllhs.f,4755: Local name CDUM1 was referenced but never set
Error #171 snllhs.f,4755: Local name CDUM2 was referenced but never set

Warning #72 snllhs.f,4755: common /SAMP/ was not used in function RDPAR()
Warning #103 snllhs.f,4943: expecting REAL constant in DATA - not CHARACTER
Warning #72 snllhs.f,4967: common /SAMP/ was not used in function SETDEF()
Warning #719 snllhs.f,5077: expecting array (to set) - string constant passed in arg #1 of XE
Warning #684 snllhs.f,5077: mixed strings and arithmetic objects in arg #1 of XERROR()
Warning #713 snllhs.f,5077: expecting lvalue as scalar argument - constant/expr passed in arg
Warning #713 snllhs.f,5077: expecting lvalue as scalar argument - constant/expr passed in arg
Warning #713 snllhs.f,5077: expecting lvalue as scalar argument - constant/expr passed in arg
Warning #719 snllhs.f,5079: expecting array (to set) - string constant passed in arg #1 of XE
Warning #684 snllhs.f,5079: mixed strings and arithmetic objects in arg #1 of XERROR()
Warning #713 snllhs.f,5079: expecting lvalue as scalar argument - constant/expr passed in arg
Warning #713 snllhs.f,5079: expecting lvalue as scalar argument - constant/expr passed in arg
Warning #713 snllhs.f,5079: expecting lvalue as scalar argument - constant/expr passed in arg
Warning #72 snllhs.f,5656: common /SAMP/ was not used in function VIF()
Warning #72 snllhs.f,5706: common /SAMP/ was not used in function WRTCRD()
Warning #72 snllhs.f,5814: common /SAMP/ was not used in function WRTPAR()
Warning #684 snllhs.f,6095: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,6096: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,6098: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,6100: mixed strings and arithmetic objects in arg #1 of XERSAV()
Warning #684 snllhs.f,6101: mixed strings and arithmetic objects in arg #1 of XERABT()
Warning #684 snllhs.f,6126: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,6129: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,6130: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,6132: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,6133: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,6166: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,6168: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,6170: mixed strings and arithmetic objects in arg #1 of XERSAV()

Warning #686 exec.f,406: different type and size used in arg #3 of EBSFAIL()
Warning #626 exec.f,409: differing number of EBSFAIL() arguments; intrf. vs. call
Warning #701 condxyzt.f,82: function TEMPGL() return type differs with interface
Warning #686 condxyzt.f,82: different type and size used in arg #1 of QGAUS()
Warning #626 ebsfail.f,657: differing number of EBSFAIL() arguments; intrf. vs. definition
Warning #702 ebsfail.f,657: routine EBSFAIL() argument definitions differ with interface
Warning #703 ebsfail.f,657: routine EBSFAIL() arg. type definitions differ with interface
Error #3 getdoc.f,6: F77 main previously detected
Error #4 nfenv.f,2: function/subr NFENV() previously defined
Error #60 nfenv.f,537: common /NFENV2/ size differs with prior use (192 vs 16000 prior)
Error #61 nfenv.f,537: common /NFENV2/ size differs with first usage (192 vs 16000 1st)
Error #60 nfenv.f,537: common /NFENV3/ size differs with prior use (192 vs 16000 prior)
Error #61 nfenv.f,537: common /NFENV3/ size differs with first usage (192 vs 16000 1st)
Error #60 nfenv.f,537: common /NFENV4/ size differs with prior use (8 vs 16000 prior)
Error #61 nfenv.f,537: common /NFENV4/ size differs with first usage (8 vs 16000 1st)
Error #60 nfenv.f,537: common /NFENV5/ size differs with prior use (4 vs 16000 prior)
Error #61 nfenv.f,537: common /NFENV5/ size differs with first usage (4 vs 16000 1st)
Error #60 nfenv.f,537: common /NFENV7/ size differs with prior use (4 vs 32 prior)
Error #61 nfenv.f,537: common /NFENV7/ size differs with first usage (4 vs 32 1st)
Error #60 nfenv.f,537: common /NFENV8/ size differs with prior use (4 vs 32 prior)
Error #61 nfenv.f,537: common /NFENV8/ size differs with first usage (4 vs 32 1st)
Warning #701 ran.f,830: function BETASPECIAL() return type differs with interface
Warning #686 ran.f,830: different type and size used in arg #1 of QGAUS()
Error #674 reader.f,1011: specific intrinsic IDINT() definition disagrees w/arg type: INT4
Warning #701 subarea.f,741: function FCNXY() return type differs with interface
Warning #686 subarea.f,741: different type and size used in arg #8 of ROOT()
Warning #626 uzflow.f,590: differing number of YUTOK_IN_USTR() arguments; intrf. vs. call
Warning #103 uzft.f,3100: expecting REAL*8 constant in DATA - not INTEGER*4

```
Warning #700 exec.f,100: No interface for routine TIME()
Warning #700 exec.f,101: No interface for routine CTIME()
Warning #718 exec.f,116: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,136: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,156: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,183: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,187: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,191: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,219: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,220: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,224: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,225: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,229: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,230: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,258: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,259: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,269: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,270: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,274: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,275: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,285: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,286: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,290: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,291: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,304: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,305: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,315: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,316: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,320: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,321: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,331: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,332: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #700 exec.f,346: No interface for routine SH()
Warning #718 exec.f,355: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,415: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #717 exec.f,491: expecting multi-dimension array - array element passed in arg #5 of
Warning #717 exec.f,545: expecting multi-dimension array - array element passed in arg #7 of
Warning #717 exec.f,606: expecting multi-dimension array - array element passed in arg #16 of
Warning #717 exec.f,664: expecting multi-dimension array - array element passed in arg #9 of
Warning #717 exec.f,666: expecting multi-dimension array - array element passed in arg #11 of
Warning #717 exec.f,728: expecting multi-dimension array - array element passed in arg #9 of
Warning #717 exec.f,730: expecting multi-dimension array - array element passed in arg #11 of
Warning #717 exec.f,813: expecting multi-dimension array - array element passed in arg #8 of
Warning #717 exec.f,814: expecting multi-dimension array - array element passed in arg #9 of
Warning #717 exec.f,855: expecting multi-dimension array - array element passed in arg #9 of
Warning #717 exec.f,858: expecting multi-dimension array - array element passed in arg #6 of
Warning #717 exec.f,869: expecting multi-dimension array - array element passed in arg #7 of
Warning #717 exec.f,888: expecting multi-dimension array - array element passed in arg #6 of
Warning #717 exec.f,907: expecting multi-dimension array - array element passed in arg #4 of
Warning #717 exec.f,912: expecting multi-dimension array - array element passed in arg #7 of
Warning #717 exec.f,913: expecting multi-dimension array - array element passed in arg #8 of
Warning #717 exec.f,916: expecting multi-dimension array - array element passed in arg #6 of
Warning #717 exec.f,923: expecting multi-dimension array - array element passed in arg #6 of
Warning #717 exec.f,938: expecting multi-dimension array - array element passed in arg #6 of
Warning #717 exec.f,942: expecting multi-dimension array - array element passed in arg #6 of
Warning #717 exec.f,948: expecting multi-dimension array - array element passed in arg #6 of
Warning #717 exec.f,952: expecting multi-dimension array - array element passed in arg #6 of
Error #171 exec.f,961: Local name TEMPREP was referenced but never set
Error #171 exec.f,961: Local name TEMPSPF was referenced but never set
Warning #718 exec.f,1092: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 exec.f,1097: expecting array - variable passed in arg #2 of CLEARCHAR()
Error #171 exec.f,1129: Local name IUNIT was referenced but never set
Warning #700 array.f,321: No interface for routine DERF()
Warning #718 ashplumo.f,18: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #543 ashplumo.f,20: loss of precision in assignment: REAL*8 to INTEGER*4
Warning #718 ashplumo.f,27: expecting array - variable passed in arg #2 of CLEARCHAR()
```

[illegible]

Warning #718 ashrmovo.f,287: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 ashrmovo.f,291: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 ashrmovo.f,295: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 ashrmovo.f,299: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 ashrmovo.f,303: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 ashrmovo.f,307: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 ashrmovo.f,311: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 ashrmovo.f,315: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 ashrmovo.f,319: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 ashrmovo.f,323: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 ashrmovo.f,327: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 ashrmovo.f,331: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 ashrmovo.f,335: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 ashrmovo.f,339: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 ashrmovo.f,343: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 ashrmovo.f,347: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 ashrmovo.f,351: expecting array - variable passed in arg #2 of CLEARCHAR()
Warning #718 ashrmovo.f,355: expecting array - variable passed in arg #2 of CLEARCHAR()

32/95

Example input file, supplied with TPA Version 3.0 Code
Supplied to NRC on March 17, 1997
TPA 3.0, Job started: Mon Mar 17 04:34:27 1997
Summed Normalized Release over 10,000 years
Realization EPA_Release_Limit
1 0.00000E+00
2 0.00000E+00
3 0.00000E+00

STANDARD Header
for TPA Version 3.0
demonstrating
Unique Run Identification.

SEW 4/3/97

pcl4

JOB 866

QAtesting

*File names for
TAA Version 3.0*

For: root
Date: Thu Apr 3 12:07:13 CST 1997

Submit queue: Ethernet
Submitted: 530:34:22
Started: 530:34:22



QMS 1725 Print System

QMS 1725 (1st floor)

| | | |
|--------------------|-------------------------|---|
| rw-r--r--x1121/100 | 0 Apr 1 15:27 1997 | QAtesting/ |
| rw-r--r--x1121/100 | 0 Mar 8 15:10 1997 | QAtesting/RAN/ |
| rw-r--r--1121/100 | 11155 Jan 31 09:38 1997 | QAtesting/RAN/testranran1.out |
| rw-r--r--1121/100 | 3665 Jan 28 09:49 1997 | QAtesting/RAN/TESTREAD |
| rw-r--r--1121/100 | 3241 Feb 6 09:35 1997 | QAtesting/RAN/RAN_README |
| rw-r--r--1121/100 | 7434 Jan 30 16:23 1997 | QAtesting/RAN/testranbeta.f |
| rw-r--r--1121/100 | 3941 Jan 30 16:26 1997 | QAtesting/RAN/testranbeta.out |
| rw-r--r--1121/100 | 6233 Jan 30 16:23 1997 | QAtesting/RAN/testranexponential.f |
| rw-r--r--1121/100 | 1835 Jan 30 16:30 1997 | QAtesting/RAN/testranexponential.out |
| rw-r--r--1121/100 | 6573 Jan 30 16:38 1997 | QAtesting/RAN/testranfiniteexponential.f |
| rw-r--r--1121/100 | 2042 Jan 30 16:40 1997 | QAtesting/RAN/testranfiniteexponential.out |
| rw-r--r--1121/100 | 7693 Jan 30 16:47 1997 | QAtesting/RAN/testrangasdev.f |
| rw-r--r--1121/100 | 3341 Jan 30 16:49 1997 | QAtesting/RAN/testrangasdev.out |
| rw-r--r--1121/100 | 3749 Jan 31 08:35 1997 | QAtesting/RAN/testranintegeruniform.f |
| rw-r--r--1121/100 | 2522 Jan 31 08:36 1997 | QAtesting/RAN/testranintegeruniform.out |
| rw-r--r--1121/100 | 7383 Jan 31 08:46 1997 | QAtesting/RAN/testranlogbeta.f |
| rw-r--r--1121/100 | 3909 Jan 31 08:50 1997 | QAtesting/RAN/testranlogbeta.out |
| rw-r--r--1121/100 | 10121 Jan 31 09:06 1997 | QAtesting/RAN/testranlognormal.f |
| rw-r--r--1121/100 | 5060 Jan 31 09:07 1997 | QAtesting/RAN/testranlognormal.out |
| rw-r--r--1121/100 | 5296 Jan 31 09:16 1997 | QAtesting/RAN/testranlogtriangular.f |
| rw-r--r--1121/100 | 9590 Jan 31 09:18 1997 | QAtesting/RAN/testranlogtriangular.out |
| rw-r--r--1121/100 | 5586 Jan 31 09:29 1997 | QAtesting/RAN/testranloguniform.f |
| rw-r--r--1121/100 | 11452 Jan 31 09:30 1997 | QAtesting/RAN/testranloguniform.out |
| rw-r--r--1121/100 | 7839 Jan 31 13:41 1997 | QAtesting/RAN/testrannormal.f |
| rw-r--r--1121/100 | 3407 Jan 31 13:42 1997 | QAtesting/RAN/testrannormal.out |
| rw-r--r--1121/100 | 3932 Feb 26 07:14 1997 | QAtesting/RAN/testranquantn.f |
| rw-r--r--1121/100 | 1356 Jan 31 10:58 1997 | QAtesting/RAN/testranquantu.f |
| rw-r--r--1121/100 | 1944 Jan 28 08:35 1997 | QAtesting/RAN/README_MATHCAD |
| rw-r--r--1121/100 | 4486 Jan 31 09:37 1997 | QAtesting/RAN/testranran1.f |
| rw-r--r--1121/100 | 5071 Jan 31 09:45 1997 | QAtesting/RAN/testrantriangular.f |
| rw-r--r--1121/100 | 13050 Jan 31 09:46 1997 | QAtesting/RAN/testrantriangular.out |
| rw-r--r--1121/100 | 5319 Jan 31 10:05 1997 | QAtesting/RAN/testranuniform.f |
| rw-r--r--1121/100 | 11439 Jan 31 10:06 1997 | QAtesting/RAN/testranuniform.out |
| rw-r--r--x1121/100 | 1857 Jan 24 14:07 1997 | QAtesting/RAN/runallrantests |
| rw-r--r--1121/100 | 13184 Mar 8 15:08 1997 | QAtesting/RAN/TESTRAN_README |
| rw-r--r--1121/100 | 942 Feb 22 08:55 1997 | QAtesting/RAN/testquantset.f |
| rw-r--r--1121/100 | 4256 Feb 25 13:49 1997 | QAtesting/RAN/testranset.f |
| rw-r--r--1121/100 | 829 Feb 25 12:53 1997 | QAtesting/RAN/testset.f |
| rw-r--r--1121/100 | 31074 Jan 15 16:40 1997 | QAtesting/RAN/testranquant.f |
| rw-rw-rw-1121/100 | 1667 Jan 31 11:17 1997 | QAtesting/RAN/testquantb.f |
| rw-rw-rw-1121/100 | 717 Jan 31 12:53 1997 | QAtesting/RAN/testquante.f |
| rw-rw-rw-1121/100 | 879 Jan 31 12:54 1997 | QAtesting/RAN/testquantfe.f |
| rw-r--r--1121/100 | 1394 Jan 30 15:04 1997 | QAtesting/RAN/testtol.f |
| rw-r--r--1121/100 | 29892 Mar 8 14:38 1997 | QAtesting/RAN/RANDOC_README |
| rw-r--r--1121/100 | 43610 Jan 27 09:25 1997 | QAtesting/RAN/testranquant.crash.f |
| rw-r--r--1121/100 | 31220 Jan 15 17:32 1997 | QAtesting/RAN/testranquant.crash.pneg.f |
| rw-rw-rw-1121/100 | 937 Jan 31 12:56 1997 | QAtesting/RAN/testquantln.f |
| rw-r--r--1121/100 | 13538 Jan 15 17:35 1997 | QAtesting/RAN/testranquant.crash.pneg.out |
| rw-r--r--1121/100 | 31212 Jan 15 17:42 1997 | QAtesting/RAN/testranquant.crash.poverone.f |
| rw-rw-rw-1121/100 | 867 Jan 31 12:58 1997 | QAtesting/RAN/testquantlt.f |
| rw-r--r--1121/100 | 13490 Jan 15 17:44 1997 | QAtesting/RAN/testranquant.crash.poverone.out |
| rw-r--r--1121/100 | 4441 Jan 31 11:02 1997 | QAtesting/RAN/testranquantlt.f |
| rw-rw-rw-1121/100 | 795 Jan 31 12:59 1997 | QAtesting/RAN/testquantlu.f |
| rw-r--r--1121/100 | 35034 Jan 27 09:29 1997 | QAtesting/RAN/testranquant.crash.out |
| rw-rw-rw-1121/100 | 855 Jan 31 13:03 1997 | QAtesting/RAN/testquanttt.f |
| rw-r--r--x1121/100 | 766 Jan 24 11:47 1997 | QAtesting/RAN/runallrandyranests |
| rw-r--r--1121/100 | 1276 Jan 31 11:20 1997 | QAtesting/RAN/testquantb.out |
| rw-r--r--1121/100 | 558 Jan 31 12:53 1997 | QAtesting/RAN/testquante.out |
| rw-r--r--1121/100 | 631 Jan 31 12:55 1997 | QAtesting/RAN/testquantfe.out |
| rw-r--r--1121/100 | 645 Jan 31 12:57 1997 | QAtesting/RAN/testquantln.out |

| | | | | | | |
|-------------------|-------|-----|----|-------|------|------------------------------------|
| rw-r--r--1121/100 | 630 | Jan | 31 | 12:59 | 1997 | Testing/RAN/testquantlt.out |
| rw-r--r--1121/100 | 583 | Jan | 31 | 13:00 | 1997 | QAtesting/RAN/testquantlu.out |
| rw-r--r--1121/100 | 596 | Jan | 31 | 13:04 | 1997 | QAtesting/RAN/testquanttt.out |
| rw-r--r--1121/100 | 21563 | Jan | 24 | 16:35 | 1997 | QAtesting/RAN/testrancrash.f |
| rw-r--r--1121/100 | 14567 | Jan | 27 | 09:13 | 1997 | QAtesting/RAN/testrancrash.out |
| rw-r--r--1121/100 | 787 | Jan | 31 | 13:05 | 1997 | QAtesting/RAN/testquantu.f |
| rw-r--r--1121/100 | 596 | Jan | 31 | 13:07 | 1997 | QAtesting/RAN/testquantu.out |
| rw-r--r--1121/100 | 3837 | Jan | 31 | 11:07 | 1997 | QAtesting/RAN/testranquantb.f |
| rw-r--r--1121/100 | 3639 | Jan | 31 | 10:38 | 1997 | QAtesting/RAN/testranquantlb.f |
| rw-r--r--1121/100 | 3443 | Jan | 31 | 10:56 | 1997 | QAtesting/RAN/testranquanttt.f |
| rw-r--r--1121/100 | 5410 | Jan | 31 | 11:04 | 1997 | QAtesting/RAN/testranquantln.f |
| rw-r--r--1121/100 | 4322 | Jan | 31 | 11:00 | 1997 | QAtesting/RAN/testranquantlu.f |
| rw-r--r--1121/100 | 4584 | Jan | 31 | 11:05 | 1997 | QAtesting/RAN/testranquante.f |
| rw-r--r--1121/100 | 5000 | Jan | 31 | 10:53 | 1997 | QAtesting/RAN/testranquantfe.f |
| rw-r--r--1121/100 | 1277 | Jan | 27 | 14:09 | 1997 | QAtesting/RAN/runallquanttests |
| rw-r--r--1121/100 | 2396 | Feb | 25 | 13:52 | 1997 | QAtesting/RAN/testranset.out |
| rw-r--r--1121/100 | 388 | Feb | 25 | 13:52 | 1997 | QAtesting/RAN/testset.out |
| rw-r--r--1121/100 | 1620 | Feb | 25 | 14:05 | 1997 | QAtesting/RAN/testquantset.out |
| rw-r--r--1121/100 | 1409 | Jan | 31 | 11:01 | 1997 | QAtesting/RAN/testranquantlu.out |
| rw-r--r--1121/100 | 7688 | Mar | 5 | 09:25 | 1997 | QAtesting/RAN/testranquantl1.out |
| rw-r--r--1121/100 | 1740 | Jan | 31 | 11:04 | 1997 | QAtesting/RAN/testranquantln.out |
| rw-r--r--1121/100 | 1307 | Jan | 31 | 11:08 | 1997 | QAtesting/RAN/testranquantb.out |
| rw-r--r--1121/100 | 1259 | Jan | 31 | 10:39 | 1997 | QAtesting/RAN/testranquantlb.out |
| rw-r--r--1121/100 | 1169 | Jan | 31 | 10:57 | 1997 | QAtesting/RAN/testranquanttt.out |
| rw-r--r--1121/100 | 1463 | Jan | 31 | 11:03 | 1997 | QAtesting/RAN/testranquantlt.out |
| rw-r--r--1121/100 | 1508 | Jan | 31 | 11:06 | 1997 | QAtesting/RAN/testranquante.out |
| rw-r--r--1121/100 | 1437 | Jan | 31 | 10:53 | 1997 | QAtesting/RAN/testranquantfe.out |
| rw-r--r--1121/100 | 8863 | Jan | 31 | 10:59 | 1997 | QAtesting/RAN/testranquantu.out |
| rw-r--r--1121/100 | 1464 | Feb | 26 | 07:47 | 1997 | QAtesting/RAN/testranquantn.out |
| rw-r--r--1121/100 | 572 | Mar | 4 | 15:21 | 1997 | QAtesting/RAN/testquantl.f |
| rw-r--r--1121/100 | 2784 | Mar | 5 | 07:47 | 1997 | QAtesting/RAN/testranquantl1.f |
| rw-r--r--1121/100 | 3287 | Mar | 5 | 16:23 | 1997 | QAtesting/RAN/testranseed.f |
| rw-r--r--1121/100 | 8238 | Mar | 5 | 09:25 | 1997 | QAtesting/RAN/testranquantl2.out |
| rw-r--r--1121/100 | 3803 | Mar | 5 | 07:48 | 1997 | QAtesting/RAN/testranquantl2.f |
| rw-r--r--1121/100 | 2996 | Mar | 5 | 11:54 | 1997 | QAtesting/RAN/testranquantset2.f |
| rw-r--r--1121/100 | 8099 | Mar | 5 | 09:26 | 1997 | QAtesting/RAN/testranquantl3.out |
| rw-r--r--1121/100 | 2804 | Mar | 5 | 11:41 | 1997 | QAtesting/RAN/testranquantset1.f |
| rw-r--r--1121/100 | 8859 | Mar | 5 | 09:27 | 1997 | QAtesting/RAN/testranquantl4.out |
| rw-r--r--1121/100 | 2967 | Mar | 5 | 07:46 | 1997 | QAtesting/RAN/testranquantl3.f |
| rw-r--r--1121/100 | 3990 | Mar | 5 | 08:06 | 1997 | QAtesting/RAN/testranquantl4.f |
| rw-r--r--1121/100 | 2528 | Mar | 5 | 14:55 | 1997 | QAtesting/RAN/testrancdfn1.f |
| rw-r--r--1121/100 | 4468 | Mar | 8 | 09:54 | 1997 | QAtesting/RAN/testranquantl10.f |
| rw-r--r--1121/100 | 2185 | Mar | 5 | 16:24 | 1997 | QAtesting/RAN/testranseed.out |
| rw-r--r--1121/100 | 8091 | Mar | 5 | 09:27 | 1997 | QAtesting/RAN/testranquantl5.out |
| rw-r--r--1121/100 | 4367 | Feb | 3 | 17:25 | 1997 | QAtesting/RAN/testiquantu.f |
| rw-r--r--1121/100 | 2711 | Feb | 3 | 17:26 | 1997 | QAtesting/RAN/testiquantu.out |
| rw-r--r--1121/100 | 2762 | Mar | 5 | 08:10 | 1997 | QAtesting/RAN/testranquantl5.f |
| rw-r--r--1121/100 | 3019 | Mar | 5 | 11:51 | 1997 | QAtesting/RAN/testranquantset3.f |
| rw-r--r--1121/100 | 3780 | Mar | 5 | 08:17 | 1997 | QAtesting/RAN/testranquantl6.f |
| rw-r--r--1121/100 | 8600 | Mar | 5 | 09:28 | 1997 | QAtesting/RAN/testranquantl6.out |
| rw-r--r--1121/100 | 8467 | Mar | 5 | 14:56 | 1997 | QAtesting/RAN/testrancdfn1.out |
| rw-r--r--1121/100 | 8600 | Mar | 5 | 08:18 | 1997 | QAtesting/RAN/testranquantl6.out |
| rw-r--r--1121/100 | 2489 | Mar | 5 | 08:28 | 1997 | QAtesting/RAN/testranquantl7.f |
| rw-r--r--1121/100 | 3230 | Mar | 5 | 12:00 | 1997 | QAtesting/RAN/testranquantset4.f |
| rw-r--r--1121/100 | 423 | Mar | 5 | 09:28 | 1997 | QAtesting/RAN/testranquantl7.out |
| rw-r--r--1121/100 | 3074 | Mar | 5 | 08:34 | 1997 | QAtesting/RAN/testranquantl8.f |
| rw-r--r--1121/100 | 2526 | Mar | 5 | 14:54 | 1997 | QAtesting/RAN/testrancdfn2.f |
| rw-r--r--1121/100 | 8114 | Mar | 5 | 09:29 | 1997 | QAtesting/RAN/testranquantl8.out |
| rw-r--r--1121/100 | 4090 | Mar | 5 | 08:42 | 1997 | QAtesting/RAN/testranquantl9.f |
| rw-r--r--1121/100 | 8354 | Mar | 5 | 11:43 | 1997 | QAtesting/RAN/testranquantset1.out |

| | | | | | | |
|-------------------|-------|-----|----|-------|------|---|
| rw-r--r--1121/100 | 9128 | Mar | 5 | 11:55 | 1997 | testing/RAN/testranquantset2.out |
| rw-r--r--1121/100 | 8937 | Mar | 5 | 09:31 | 1997 | QAtesting/RAN/testranquantl9.out |
| rw-r--r--1121/100 | 8866 | Mar | 5 | 11:52 | 1997 | QAtesting/RAN/testranquantset3.out |
| rw-r--r--1121/100 | 8981 | Mar | 5 | 12:01 | 1997 | QAtesting/RAN/testranquantset4.out |
| rw-r--r--1121/100 | 2829 | Mar | 5 | 12:03 | 1997 | QAtesting/RAN/testranquantset5.f |
| rw-r--r--1121/100 | 8352 | Mar | 5 | 12:04 | 1997 | QAtesting/RAN/testranquantset5.out |
| rw-r--r--1121/100 | 3044 | Mar | 5 | 13:00 | 1997 | QAtesting/RAN/testranquantset6.f |
| rw-r--r--1121/100 | 8188 | Mar | 5 | 14:54 | 1997 | QAtesting/RAN/testrancdfn2.out |
| rw-r--r--1121/100 | 9133 | Mar | 5 | 13:10 | 1997 | QAtesting/RAN/testranquantset6.out |
| rw-r--r--1121/100 | 2718 | Mar | 5 | 13:13 | 1997 | QAtesting/RAN/testranquantset7.f |
| rw-r--r--1121/100 | 4405 | Mar | 5 | 15:29 | 1997 | QAtesting/RAN/testrancdfn3.f |
| rw-r--r--1121/100 | 8235 | Mar | 5 | 13:14 | 1997 | QAtesting/RAN/testranquantset7.out |
| rw-r--r--1121/100 | 2696 | Mar | 5 | 13:17 | 1997 | QAtesting/RAN/testranquantset8.f |
| rw-r--r--1121/100 | 966 | Mar | 5 | 13:18 | 1997 | QAtesting/RAN/testranquantset8.out |
| rw-r--r--1121/100 | 1666 | Mar | 5 | 15:30 | 1997 | QAtesting/RAN/testrancdfn3.out |
| rw-r--r--1121/100 | 1259 | Mar | 5 | 15:38 | 1997 | QAtesting/RAN/testrancdfn4.f |
| rw-r--r--1121/100 | 523 | Mar | 5 | 15:38 | 1997 | QAtesting/RAN/testrancdfn4.out |
| rw-r--r--1121/100 | 1220 | Mar | 5 | 15:39 | 1997 | QAtesting/RAN/testrancdfn5.f |
| rw-r--r--1121/100 | 502 | Mar | 5 | 15:39 | 1997 | QAtesting/RAN/testrancdfn5.out |
| rw-r--r--1121/100 | 2820 | Mar | 8 | 09:54 | 1997 | QAtesting/RAN/testranquantl10.out |
| rw-r--r--1121/100 | 0 | Mar | 8 | 12:20 | 1997 | QAtesting/INVENT/ |
| rw-r--r--1121/100 | 1816 | Jan | 27 | 15:29 | 1997 | QAtesting/INVENT/testinvent.problem34.f |
| rw-r--r--1121/100 | 8115 | Mar | 8 | 12:17 | 1997 | QAtesting/INVENT/TESTINVENT_README |
| rw-r--r--1121/100 | 12520 | Mar | 7 | 09:28 | 1997 | QAtesting/INVENT/testinventqpermtu.f |
| rw-r--r--1121/100 | 34701 | Mar | 8 | 12:22 | 1997 | QAtesting/INVENT/INVENTDOC_README |
| rw-r--r--1121/100 | 6430 | Feb | 5 | 16:31 | 1997 | QAtesting/INVENT/INVENT_README |
| rw-r--r--1121/100 | 1687 | Jan | 14 | 11:36 | 1997 | QAtesting/INVENT/testinvent.problem10.f |
| rw-r--r--1121/100 | 9467 | Jan | 14 | 11:36 | 1997 | QAtesting/INVENT/testinvent.problem10.out |
| rw-r--r--1121/100 | 3241 | Jan | 27 | 10:28 | 1997 | QAtesting/INVENT/testinvent.problem11.f |
| rw-r--r--1121/100 | 4948 | Jan | 27 | 10:31 | 1997 | QAtesting/INVENT/testinvent.problem11.out |
| rw-r--r--1121/100 | 3237 | Jan | 27 | 10:33 | 1997 | QAtesting/INVENT/testinvent.problem12.f |
| rw-r--r--1121/100 | 4944 | Jan | 27 | 10:36 | 1997 | QAtesting/INVENT/testinvent.problem12.out |
| rw-r--r--1121/100 | 3312 | Jan | 27 | 10:39 | 1997 | QAtesting/INVENT/testinvent.problem13.f |
| rw-r--r--1121/100 | 5496 | Jan | 27 | 10:42 | 1997 | QAtesting/INVENT/testinvent.problem13.out |
| rw-r--r--1121/100 | 3308 | Jan | 27 | 10:45 | 1997 | QAtesting/INVENT/testinvent.problem14.f |
| rw-r--r--1121/100 | 5495 | Jan | 27 | 10:47 | 1997 | QAtesting/INVENT/testinvent.problem14.out |
| rw-r--r--1121/100 | 1835 | Jan | 27 | 10:49 | 1997 | QAtesting/INVENT/testinvent.problem15.f |
| rw-r--r--1121/100 | 3613 | Jan | 27 | 10:52 | 1997 | QAtesting/INVENT/testinvent.problem15.out |
| rw-r--r--1121/100 | 1831 | Jan | 27 | 10:53 | 1997 | QAtesting/INVENT/testinvent.problem16.f |
| rw-r--r--1121/100 | 3602 | Jan | 27 | 10:55 | 1997 | QAtesting/INVENT/testinvent.problem16.out |
| rw-r--r--1121/100 | 6486 | Jan | 29 | 13:38 | 1997 | QAtesting/INVENT/testinvent.problem17.f |
| rw-r--r--1121/100 | 4598 | Jan | 27 | 11:07 | 1997 | QAtesting/INVENT/testinvent.problem17.out |
| rw-r--r--1121/100 | 6482 | Jan | 29 | 13:40 | 1997 | QAtesting/INVENT/testinvent.problem18.f |
| rw-r--r--1121/100 | 4585 | Jan | 27 | 11:09 | 1997 | QAtesting/INVENT/testinvent.problem18.out |
| rw-r--r--1121/100 | 3253 | Jan | 27 | 11:15 | 1997 | QAtesting/INVENT/testinvent.problem19.f |
| rw-r--r--1121/100 | 4969 | Jan | 27 | 12:43 | 1997 | QAtesting/INVENT/testinvent.problem19.out |
| rw-r--r--1121/100 | 6834 | Feb | 19 | 16:18 | 1997 | QAtesting/INVENT/testinvent.problem1a.f |
| rw-r--r--1121/100 | 5528 | Jan | 24 | 16:58 | 1997 | QAtesting/INVENT/testinvent.problem1a.out |
| rw-r--r--1121/100 | 6765 | Jan | 14 | 11:36 | 1997 | QAtesting/INVENT/testinvent.problem1b.f |
| rw-r--r--1121/100 | 5423 | Jan | 14 | 11:36 | 1997 | QAtesting/INVENT/testinvent.problem1b.out |
| rw-r--r--1121/100 | 3249 | Jan | 27 | 12:45 | 1997 | QAtesting/INVENT/testinvent.problem20.f |
| rw-r--r--1121/100 | 4972 | Jan | 27 | 12:49 | 1997 | QAtesting/INVENT/testinvent.problem20.out |
| rw-r--r--1121/100 | 3405 | Jan | 27 | 14:33 | 1997 | QAtesting/INVENT/testinvent.problem21.f |
| rw-r--r--1121/100 | 4747 | Jan | 27 | 14:33 | 1997 | QAtesting/INVENT/testinvent.problem21.out |
| rw-r--r--1121/100 | 3401 | Jan | 27 | 14:33 | 1997 | QAtesting/INVENT/testinvent.problem22.f |
| rw-r--r--1121/100 | 4744 | Jan | 27 | 14:34 | 1997 | QAtesting/INVENT/testinvent.problem22.out |
| rw-r--r--1121/100 | 5088 | Jan | 27 | 13:21 | 1997 | QAtesting/INVENT/testinvent.problem23.f |
| rw-r--r--1121/100 | 5674 | Jan | 27 | 13:15 | 1997 | QAtesting/INVENT/testinvent.problem23.out |
| rw-r--r--1121/100 | 5084 | Jan | 27 | 13:07 | 1997 | QAtesting/INVENT/testinvent.problem24.f |
| rw-r--r--1121/100 | 5669 | Jan | 27 | 13:10 | 1997 | QAtesting/INVENT/testinvent.problem24.out |

```

rw-r--r--1121/100 8302 Jan 27 14:36 1997 .esting/INVENT/testinvent.problem25.f
rw-r--r--1121/100 4666 Jan 14 11:36 1997 QAtesting/INVENT/testinvent.problem25.out
rw-r--r--1121/100 5593 Jan 27 13:29 1997 QAtesting/INVENT/testinvent.problem26.f
rw-r--r--1121/100 2587 Jan 27 13:32 1997 QAtesting/INVENT/testinvent.problem26.out
rw-r--r--1121/100 6866 Jan 27 13:38 1997 QAtesting/INVENT/testinvent.problem27.f
rw-r--r--1121/100 7790 Jan 27 13:37 1997 QAtesting/INVENT/testinvent.problem27.out
rw-r--r--1121/100 6862 Jan 27 13:39 1997 QAtesting/INVENT/testinvent.problem28.f
rw-r--r--1121/100 7789 Jan 27 13:41 1997 QAtesting/INVENT/testinvent.problem28.out
rw-r--r--1121/100 11224 Jan 27 13:44 1997 QAtesting/INVENT/testinvent.problem29.f
rw-r--r--1121/100 10988 Jan 27 13:47 1997 QAtesting/INVENT/testinvent.problem29.out
rw-r--r--1121/100 10840 Jan 14 11:36 1997 QAtesting/INVENT/testinvent.problem2a.f
rw-r--r--1121/100 9925 Jan 14 11:36 1997 QAtesting/INVENT/testinvent.problem2a.out
rw-r--r--1121/100 10799 Jan 14 11:36 1997 QAtesting/INVENT/testinvent.problem2b.f
rw-r--r--1121/100 9894 Jan 14 11:36 1997 QAtesting/INVENT/testinvent.problem2b.out
rw-r--r--1121/100 3691 Jan 14 11:36 1997 QAtesting/INVENT/testinvent.problem30.f
rw-r--r--1121/100 6858 Jan 14 11:36 1997 QAtesting/INVENT/testinvent.problem30.out
rw-r--r--1121/100 5221 Jan 14 11:36 1997 QAtesting/INVENT/testinvent.problem31.f
rw-r--r--1121/100 9923 Jan 14 11:36 1997 QAtesting/INVENT/testinvent.problem31.out
rw-r--r--1121/100 3815 Jan 14 11:36 1997 QAtesting/INVENT/testinvent.problem32.f
rw-r--r--1121/100 1940 Jan 14 11:36 1997 QAtesting/INVENT/testinvent.problem32.out
rw-r--r--1121/100 15122 Jan 14 11:36 1997 QAtesting/INVENT/testinvent.problem33.f
rw-r--r--1121/100 9450 Jan 14 11:36 1997 QAtesting/INVENT/testinvent.problem33.out
rw-r--r--1121/100 12126 Jan 27 13:57 1997 QAtesting/INVENT/testinvent.problem4.f
rw-r--r--1121/100 32026 Jan 27 13:58 1997 QAtesting/INVENT/testinvent.problem4.out
rw-r--r--1121/100 1870 Jan 27 14:01 1997 QAtesting/INVENT/testinvent.problem5.f
rw-r--r--1121/100 8480 Jan 27 14:04 1997 QAtesting/INVENT/testinvent.problem5.out
rw-r--r--1121/100 1866 Jan 27 14:08 1997 QAtesting/INVENT/testinvent.problem6.f
rw-r--r--1121/100 8475 Jan 27 14:10 1997 QAtesting/INVENT/testinvent.problem6.out
rw-r--r--1121/100 1779 Jan 27 10:01 1997 QAtesting/INVENT/testinvent.problem7.f
rw-r--r--1121/100 2022 Jan 27 10:15 1997 QAtesting/INVENT/testinvent.problem7.out
rw-r--r--1121/100 1778 Jan 27 14:25 1997 QAtesting/INVENT/testinvent.problem8.f
rw-r--r--1121/100 2013 Jan 27 14:27 1997 QAtesting/INVENT/testinvent.problem8.out
rw-r--r--1121/100 1544 Jan 14 11:36 1997 QAtesting/INVENT/testinvent.problem9.f
rw-r--r--1121/100 12294 Jan 14 11:36 1997 QAtesting/INVENT/testinvent.problem9.out
rw-r--r--1121/100 4283 Jan 14 11:37 1997 QAtesting/INVENT/runinventalltests
rw-r--r--1121/100 1801 Jan 27 15:33 1997 QAtesting/INVENT/testinvent.problem35.f
rw-r--r--1121/100 1864 Jan 27 15:31 1997 QAtesting/INVENT/testinvent.problem34.out
rw-r--r--1121/100 1849 Jan 27 15:35 1997 QAtesting/INVENT/testinvent.problem35.out
rw-r--r--1121/100 1394 Mar 6 17:05 1997 QAtesting/INVENT/testtol.f
rw-r--r--1121/100 9116 Mar 7 09:29 1997 QAtesting/INVENT/testinventqpermtu.out
rwxr-xr-x1121/100 0 Mar 8 17:23 1997 QAtesting/SUBAREA/
rwxr-xr-x1121/100 1949 Jan 14 11:44 1997 QAtesting/SUBAREA/runallsubareatests
rw-r--r--1121/100 3104 Feb 7 12:13 1997 QAtesting/SUBAREA/SUBAREA_README
rw-r--r--1121/100 9758 Jan 14 11:44 1997 QAtesting/SUBAREA/problem.basic.out
rw-r--r--1121/100 875 Jan 14 11:44 1997 QAtesting/SUBAREA/problem1.out
rw-r--r--1121/100 738 Jan 14 11:44 1997 QAtesting/SUBAREA/problem10.out
rw-r--r--1121/100 1162 Jan 14 11:44 1997 QAtesting/SUBAREA/problem11.out
rw-r--r--1121/100 1294 Jan 14 11:44 1997 QAtesting/SUBAREA/problem12.out
rw-r--r--1121/100 1545 Jan 14 11:44 1997 QAtesting/SUBAREA/problem13.out
rw-r--r--1121/100 1249 Jan 14 11:44 1997 QAtesting/SUBAREA/problem14.out
rw-r--r--1121/100 1264 Jan 14 11:44 1997 QAtesting/SUBAREA/problem15.out
rw-r--r--1121/100 1249 Jan 14 11:44 1997 QAtesting/SUBAREA/problem16.out
rw-r--r--1121/100 1190 Jan 14 11:44 1997 QAtesting/SUBAREA/problem18.out
rw-r--r--1121/100 2707 Jan 14 11:44 1997 QAtesting/SUBAREA/problem19.out
rw-r--r--1121/100 1161 Jan 14 11:44 1997 QAtesting/SUBAREA/problem2.out
rw-r--r--1121/100 1161 Jan 14 11:44 1997 QAtesting/SUBAREA/problem3.out
rw-r--r--1121/100 1179 Jan 14 11:44 1997 QAtesting/SUBAREA/problem4.out
rw-r--r--1121/100 1280 Jan 14 11:44 1997 QAtesting/SUBAREA/problem5.out
rw-r--r--1121/100 1213 Jan 14 11:44 1997 QAtesting/SUBAREA/problem6.out
rw-r--r--1121/100 1750 Jan 14 11:44 1997 QAtesting/SUBAREA/problem7.out

```

39/95

| | | | | | | |
|-----------|----------|-------|--------|-------|------|---|
| rw-r--r-- | 1121/100 | 1750 | Jan 14 | 11:44 | 1997 | testing/SUBAREA/problem8.out |
| rw-r--r-- | 1121/100 | 1167 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/problem9.out |
| rw-r--r-- | 1121/100 | 427 | Jan 28 | 22:10 | 1997 | QAtesting/SUBAREA/README_ERRORFUNCTION |
| rw-r--r-- | 1121/100 | 21820 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.basic.f |
| rw-r--r-- | 1121/100 | 2251 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem1.f |
| rw-r--r-- | 1121/100 | 2036 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem10.f |
| rw-r--r-- | 1121/100 | 2777 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem11.f |
| rw-r--r-- | 1121/100 | 2769 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem12.f |
| rw-r--r-- | 1121/100 | 2757 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem13.f |
| rw-r--r-- | 1121/100 | 2757 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem14.f |
| rw-r--r-- | 1121/100 | 2757 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem15.f |
| rw-r--r-- | 1121/100 | 2757 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem16.f |
| rw-r--r-- | 1121/100 | 2775 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem18.f |
| rw-r--r-- | 1121/100 | 4939 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem19.f |
| rw-r--r-- | 1121/100 | 2776 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem2.f |
| rw-r--r-- | 1121/100 | 4444 | Mar 8 | 17:22 | 1997 | QAtesting/SUBAREA/TESTSUBAREA_README |
| rw-r--r-- | 1121/100 | 2776 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem3.f |
| rw-r--r-- | 1121/100 | 2779 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem4.f |
| rw-r--r-- | 1121/100 | 2760 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem5.f |
| rw-r--r-- | 1121/100 | 2721 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem6.f |
| rw-r--r-- | 1121/100 | 2722 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem7.f |
| rw-r--r-- | 1121/100 | 2736 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem8.f |
| rw-r--r-- | 1121/100 | 2751 | Jan 14 | 11:44 | 1997 | QAtesting/SUBAREA/testsubarea.problem9.f |
| rw-r--r-- | 1121/100 | 2891 | Feb 7 | 13:48 | 1997 | QAtesting/SUBAREA/problem20.out |
| rw-r--r-- | 1121/100 | 1010 | Feb 7 | 13:42 | 1997 | QAtesting/SUBAREA/testsubarea.problem20.f |
| rw-r--r-- | 1121/100 | 18070 | Mar 8 | 17:25 | 1997 | QAtesting/SUBAREA/SUBAREADOC_README |
| rw-r--r-- | 1121/100 | 0 | Mar 8 | 10:42 | 1997 | QAtesting/ARRAY/ |
| rw-r--r-- | 1121/100 | 27110 | Mar 8 | 09:58 | 1997 | QAtesting/ARRAY/testarray.f |
| rw-r--r-- | 1121/100 | 16496 | Mar 8 | 09:59 | 1997 | QAtesting/ARRAY/testarray.out |
| rw-r--r-- | 1121/100 | 6421 | Mar 5 | 18:20 | 1997 | QAtesting/ARRAY/testarrayduplicates.out |
| rw-r--r-- | 1121/100 | 8784 | Mar 8 | 11:21 | 1997 | QAtesting/ARRAY/TESTARRAY_README |
| rw-r--r-- | 1121/100 | 19096 | Mar 8 | 11:02 | 1997 | QAtesting/ARRAY/ARRAYDOC_README |
| rw-r--r-- | 1121/100 | 7734 | Mar 5 | 18:20 | 1997 | QAtesting/ARRAY/testarrayduplicates.f |
| rw-r--r-- | 1121/100 | 7688 | Mar 5 | 18:32 | 1997 | QAtesting/ARRAY/testarrayintegerduplicates.f |
| rw-r--r-- | 1121/100 | 5330 | Mar 5 | 18:33 | 1997 | QAtesting/ARRAY/testarrayintegerduplicates.out |
| rw-r--r-- | 1121/100 | 0 | Mar 21 | 11:26 | 1997 | QAtesting/SAMPLER/ |
| rw-r--r-- | 1121/100 | 2394 | Jan 30 | 17:15 | 1997 | QAtesting/SAMPLER/testspprinttitle.f |
| rw-r--r-- | 1121/100 | 15872 | Feb 4 | 13:19 | 1997 | QAtesting/SAMPLER/testsplhs3.f |
| rw-r--r-- | 1121/100 | 784 | Feb 4 | 11:15 | 1997 | QAtesting/SAMPLER/testsplhs3.inp |
| rw-r--r-- | 1121/100 | 4954 | Feb 7 | 08:56 | 1997 | QAtesting/SAMPLER/SAMPLER_README |
| rw-r--r-- | 1121/100 | 8277 | Jan 31 | 09:55 | 1997 | QAtesting/SAMPLER/testsplhs1.out |
| rw-r--r-- | 1121/100 | 56291 | Jan 31 | 14:30 | 1997 | QAtesting/SAMPLER/testsampledparameter.out |
| rw-r--r-- | 1121/100 | 5677 | Mar 21 | 10:36 | 1997 | QAtesting/SAMPLER/TESTSAMPLER_README |
| rw-r--r-- | 1121/100 | 341 | Jan 31 | 07:32 | 1997 | QAtesting/SAMPLER/testspprinttitle.inp |
| rw-r--r-- | 1121/100 | 26065 | Mar 8 | 17:19 | 1997 | QAtesting/SAMPLER/SAMPLERDOC_README |
| rw-r--r-- | 1121/100 | 6138 | Jan 31 | 09:20 | 1997 | QAtesting/SAMPLER/testsplhs1.f |
| rw-r--r-- | 1121/100 | 7400 | Jan 30 | 13:34 | 1997 | QAtesting/SAMPLER/testhazardcurve.f |
| rw-r--r-- | 1121/100 | 337 | Feb 3 | 07:31 | 1997 | QAtesting/SAMPLER/testsplhs1.inp |
| rw-r--r-- | 1121/100 | 3844 | Jan 30 | 14:15 | 1997 | QAtesting/SAMPLER/testhazardcurve.out |
| rw-r--r-- | 1121/100 | 632 | Feb 3 | 08:38 | 1997 | QAtesting/SAMPLER/testspprintvalue.inp |
| rw-r--r-- | 1121/100 | 1219 | Jan 30 | 11:07 | 1997 | QAtesting/SAMPLER/testsampledparameter.crashalpha.out |
| rw-r--r-- | 1121/100 | 1226 | Jan 30 | 11:02 | 1997 | QAtesting/SAMPLER/testsampledparameter.crashbeta.out |
| rw-r--r-- | 1121/100 | 10953 | Jan 30 | 16:04 | 1997 | QAtesting/SAMPLER/testhazardcurvebreak.out |
| rw-r--r-- | 1121/100 | 5038 | Jan 30 | 09:55 | 1997 | QAtesting/SAMPLER/testsampledparameter.crashequalmaxminpeak.out |
| rw-r--r-- | 1121/100 | 46954 | Feb 4 | 13:19 | 1997 | QAtesting/SAMPLER/testsplhs3.out |
| rw-r--r-- | 1121/100 | 25442 | Feb 4 | 13:17 | 1997 | QAtesting/SAMPLER/testsplhs2.out |
| rw-r--r-- | 1121/100 | 3889 | Jan 30 | 09:27 | 1997 | QAtesting/SAMPLER/testsampledparameter.crashmaxminpeak.out |
| rw-r--r-- | 1121/100 | 4725 | Jan 30 | 11:04 | 1997 | QAtesting/SAMPLER/testsampledparameter.crashalpha.f |
| rw-r--r-- | 1121/100 | 7282 | Feb 4 | 14:57 | 1997 | QAtesting/SAMPLER/testsplhs4.f |
| rw-r--r-- | 1121/100 | 4733 | Jan 30 | 10:56 | 1997 | QAtesting/SAMPLER/testsampledparameter.crashbeta.f |

40/95

| | | | | | | |
|-------------------|--------|-----|----|-------|------|--|
| rw-r--r--1121/100 | 9835 | Feb | 4 | 14:58 | 1997 | ..testing/SAMPLER/testsplhs4.out |
| rw-r--r--1121/100 | 786 | Feb | 4 | 14:58 | 1997 | QAtesting/SAMPLER/testsplhs4.inp |
| rw-r--r--1121/100 | 9236 | Jan | 30 | 09:54 | 1997 | QAtesting/SAMPLER/testsampledparameter.crashequalmaxinpeak.f |
| rw-r--r--1121/100 | 2129 | Jan | 30 | 09:23 | 1997 | QAtesting/SAMPLER/testsampledparameter.crashnegatives.out |
| rw-r--r--1121/100 | 6738 | Jan | 30 | 09:25 | 1997 | QAtesting/SAMPLER/testsampledparameter.crashmaxinpeak.f |
| rw-r--r--1121/100 | 5628 | Feb | 3 | 09:22 | 1997 | QAtesting/SAMPLER/testspprintvalue.f |
| rw-r--r--1121/100 | 5314 | Jan | 30 | 09:23 | 1997 | QAtesting/SAMPLER/testsampledparameter.crashnegatives.f |
| rw-r--r--1121/100 | 3429 | Jan | 30 | 11:46 | 1997 | QAtesting/SAMPLER/testsampledparameter.crashnonewspdb.f |
| rw-r--r--1121/100 | 2578 | Jan | 30 | 11:47 | 1997 | QAtesting/SAMPLER/testsampledparameter.crashnonewspdb.out |
| rw-r--r--1121/100 | 33289 | Feb | 26 | 10:43 | 1997 | QAtesting/SAMPLER/testspcorr1.out |
| rw-r--r--1121/100 | 3253 | Feb | 27 | 13:46 | 1997 | QAtesting/SAMPLER/testspcorr1.f |
| rw-r--r--1121/100 | 57163 | Jan | 30 | 13:05 | 1997 | QAtesting/SAMPLER/testsampledparameter.f |
| rw-r--r--1121/100 | 33290 | Feb | 26 | 10:44 | 1997 | QAtesting/SAMPLER/testspcorr2.out |
| rw-r--r--1121/100 | 3254 | Feb | 26 | 09:29 | 1997 | QAtesting/SAMPLER/testspcorr3.f |
| rw-r--r--1121/100 | 3246 | Feb | 26 | 09:22 | 1997 | QAtesting/SAMPLER/testspcorr2.f |
| rw-r--r--1121/100 | 33299 | Feb | 26 | 10:44 | 1997 | QAtesting/SAMPLER/testspcorr3.out |
| rw-r--r--1121/100 | 2868 | Jan | 30 | 17:15 | 1997 | QAtesting/SAMPLER/testspprinttitle.out |
| rw-r--r--1121/100 | 64131 | Jan | 31 | 16:11 | 1997 | QAtesting/SAMPLER/testsampledparameterb.f |
| rw-r--r--1121/100 | 1394 | Jan | 31 | 08:26 | 1997 | QAtesting/SAMPLER/testtol.f |
| rw-r--r--1121/100 | 18464 | Jan | 30 | 16:03 | 1997 | QAtesting/SAMPLER/testhazardcurvebreak.f |
| rw-r--r--1121/100 | 5459 | Feb | 3 | 09:13 | 1997 | QAtesting/SAMPLER/testspprintvaluelhs.out |
| rw-r--r--1121/100 | 3336 | Feb | 26 | 09:42 | 1997 | QAtesting/SAMPLER/testspcorr4.f |
| rw-r--r--1121/100 | 245 | Feb | 26 | 10:41 | 1997 | QAtesting/SAMPLER/testspcorr1.inp |
| rw-r--r--1121/100 | 63342 | Feb | 6 | 11:02 | 1997 | QAtesting/SAMPLER/testsampledparameterb.out |
| rw-r--r--1121/100 | 7742 | Feb | 4 | 13:16 | 1997 | QAtesting/SAMPLER/testsplhs2.f |
| rw-r--r--1121/100 | 699 | Feb | 4 | 11:24 | 1997 | QAtesting/SAMPLER/testsplhs2.inp |
| rw-r--r--1121/100 | 714 | Feb | 3 | 08:51 | 1997 | QAtesting/SAMPLER/testspprintvaluelhs.inp |
| rw-r--r--1121/100 | 229 | Feb | 26 | 10:42 | 1997 | QAtesting/SAMPLER/testspcorr4.inp |
| rw-r--r--1121/100 | 6596 | Feb | 3 | 09:12 | 1997 | QAtesting/SAMPLER/testspprintvaluelhs.f |
| rw-r--r--1121/100 | 33365 | Feb | 26 | 10:45 | 1997 | QAtesting/SAMPLER/testspcorr4.out |
| rw-r--r--1121/100 | 3332 | Feb | 26 | 10:16 | 1997 | QAtesting/SAMPLER/testspcorr5.f |
| rw-r--r--1121/100 | 244 | Feb | 26 | 10:42 | 1997 | QAtesting/SAMPLER/testspcorr5.inp |
| rw-r--r--1121/100 | 4872 | Feb | 3 | 09:16 | 1997 | QAtesting/SAMPLER/testspprintvalue.out |
| rw-r--r--1121/100 | 33331 | Feb | 26 | 10:45 | 1997 | QAtesting/SAMPLER/testspcorr5.out |
| rw-r--r--1121/100 | 3401 | Feb | 26 | 10:30 | 1997 | QAtesting/SAMPLER/testspcorr6.f |
| rw-r--r--1121/100 | 311 | Feb | 26 | 10:42 | 1997 | QAtesting/SAMPLER/testspcorr6.inp |
| rw-r--r--1121/100 | 33476 | Feb | 26 | 10:46 | 1997 | QAtesting/SAMPLER/testspcorr6.out |
| rw-r--r--1121/100 | 3263 | Feb | 26 | 10:53 | 1997 | QAtesting/SAMPLER/testspcorr7.f |
| rw-r--r--1121/100 | 245 | Feb | 26 | 10:54 | 1997 | QAtesting/SAMPLER/testspcorr7.inp |
| rw-r--r--1121/100 | 33310 | Feb | 26 | 10:55 | 1997 | QAtesting/SAMPLER/testspcorr7.out |
| rw-r--r--1121/100 | 3374 | Feb | 26 | 11:22 | 1997 | QAtesting/SAMPLER/testspcorr8.f |
| rw-r--r--1121/100 | 328 | Feb | 26 | 11:22 | 1997 | QAtesting/SAMPLER/testspcorr8.inp |
| rw-r--r--1121/100 | 33511 | Feb | 26 | 11:23 | 1997 | QAtesting/SAMPLER/testspcorr8.out |
| rw-r--r--1121/100 | 334 | Feb | 26 | 11:48 | 1997 | QAtesting/SAMPLER/testspcorr9.inp |
| rw-r--r--1121/100 | 3381 | Feb | 26 | 11:48 | 1997 | QAtesting/SAMPLER/testspcorr9.f |
| rw-r--r--1121/100 | 308018 | Feb | 26 | 11:49 | 1997 | QAtesting/SAMPLER/testspcorr9.out |
| rw-r--r--1121/100 | 3337 | Feb | 27 | 16:35 | 1997 | QAtesting/SAMPLER/testspcorr10.f |
| rw-r--r--1121/100 | 216 | Feb | 26 | 10:41 | 1997 | QAtesting/SAMPLER/testspcorr2.inp |
| rw-r--r--1121/100 | 260 | Feb | 26 | 13:12 | 1997 | QAtesting/SAMPLER/testspcorr10.inp |
| rw-r--r--1121/100 | 33364 | Feb | 26 | 13:13 | 1997 | QAtesting/SAMPLER/testspcorr10.out |
| rw-r--r--1121/100 | 225 | Feb | 26 | 10:41 | 1997 | QAtesting/SAMPLER/testspcorr3.inp |
| rw-r--r--1121/100 | 3332 | Feb | 26 | 14:48 | 1997 | QAtesting/SAMPLER/testspcorr11.f |
| rw-r--r--1121/100 | 243 | Feb | 26 | 14:48 | 1997 | QAtesting/SAMPLER/testspcorr11.inp |
| rw-r--r--1121/100 | 42359 | Feb | 26 | 15:46 | 1997 | QAtesting/SAMPLER/testspcorr11.out |
| rw-r--r--1121/100 | 1708 | Mar | 17 | 14:33 | 1997 | QAtesting/SAMPLER/inpfilemget.inp |
| rw-r--r--1121/100 | 382 | Mar | 17 | 15:17 | 1997 | QAtesting/SAMPLER/inpfileiget.inp |
| rw-r--r--1121/100 | 14235 | Mar | 17 | 16:34 | 1997 | QAtesting/SAMPLER/testspmget.f |
| rw-r--r--1121/100 | 6330 | Mar | 17 | 15:37 | 1997 | QAtesting/SAMPLER/testspiget.f |
| rw-r--r--1121/100 | 1567 | Mar | 17 | 16:00 | 1997 | QAtesting/SAMPLER/inpfiledget.inp |
| rw-r--r--1121/100 | 3488 | Feb | 27 | 12:35 | 1997 | QAtesting/SAMPLER/testspcorr13.f |

rw-r--r--1121/100 288 Feb 27 12:37 1997 QAtesting/SAMPLER/testspcorr13.inp
rw-r--r--1121/100 9750 Mar 17 16:33 1997 QAtesting/SAMPLER/testspdget.f
rw-r--r--1121/100 5018 Feb 27 17:43 1997 QAtesting/SAMPLER/testspcorr1bin.f
rw-r--r--1121/100 36552 Mar 17 16:33 1997 QAtesting/SAMPLER/testspdget.out
rw-r--r--1121/100 5973 Mar 17 16:34 1997 QAtesting/SAMPLER/testspiget.out
rw-r--r--1121/100 13045 Mar 17 16:35 1997 QAtesting/SAMPLER/testspmget.out
rw-r--r--1121/100 4882 Feb 27 17:01 1997 QAtesting/SAMPLER/testspcorr1bin.out
rw-r--r--1121/100 5032 Feb 27 17:42 1997 QAtesting/SAMPLER/testspcorr2bin.f
rw-r--r--1121/100 261 Feb 26 16:00 1997 QAtesting/SAMPLER/testspcorr12.inp
rw-r--r--1121/100 3260 Feb 26 15:59 1997 QAtesting/SAMPLER/testspcorr12.f
rw-r--r--1121/100 33301 Feb 26 16:01 1997 QAtesting/SAMPLER/testspcorr12.out
rw-r--r--1121/100 5225 Feb 28 07:40 1997 QAtesting/SAMPLER/testspcorr5bin.f
rw-r--r--1121/100 4880 Feb 27 17:47 1997 QAtesting/SAMPLER/testspcorr2bin.out
rw-r--r--1121/100 5091 Feb 28 07:10 1997 QAtesting/SAMPLER/testspcorr3bin.f
rw-r--r--1121/100 4765 Feb 28 07:10 1997 QAtesting/SAMPLER/testspcorr3bin.out
rw-r--r--1121/100 5224 Feb 28 07:33 1997 QAtesting/SAMPLER/testspcorr4bin.f
rw-r--r--1121/100 4788 Feb 28 07:30 1997 QAtesting/SAMPLER/testspcorr4bin.out
rw-r--r--1121/100 4932 Feb 28 08:42 1997 QAtesting/SAMPLER/testspcorr5bin.out
rw-r--r--1121/100 5225 Feb 28 08:42 1997 QAtesting/SAMPLER/testspcorr6bin.f
rw-r--r--1121/100 5011 Feb 28 08:52 1997 QAtesting/SAMPLER/testspcorr6bin.out
rw-r--r--1121/100 5109 Feb 28 09:04 1997 QAtesting/SAMPLER/testspcorr7bin.f
rw-r--r--1121/100 4729 Feb 28 09:04 1997 QAtesting/SAMPLER/testspcorr7bin.out
rw-r--r--1121/100 5108 Feb 28 09:06 1997 QAtesting/SAMPLER/testspcorr8bin.f
rw-r--r--1121/100 4855 Feb 28 09:08 1997 QAtesting/SAMPLER/testspcorr8bin.out
rw-r--r--1121/100 5290 Feb 28 09:15 1997 QAtesting/SAMPLER/testspcorr10bin.f
rw-r--r--1121/100 4882 Feb 28 09:16 1997 QAtesting/SAMPLER/testspcorr10bin.out
rw-r--r--1121/100 5239 Feb 28 09:26 1997 QAtesting/SAMPLER/testspcorr11bin.f
rw-r--r--1121/100 4827 Feb 28 09:49 1997 QAtesting/SAMPLER/testspcorr11bin.out
rw-r--r--1121/100 5111 Feb 28 09:55 1997 QAtesting/SAMPLER/testspcorr12bin.f
rw-r--r--1121/100 4722 Feb 28 09:56 1997 QAtesting/SAMPLER/testspcorr12bin.out
rwxr-xr-x1121/100 0 Mar 31 07:28 1997 QAtesting/READER/
rw-r--r--1121/100 5168 Feb 25 07:32 1997 QAtesting/READER/testreader1.f
rw-r--r--1121/100 5311 Mar 18 15:41 1997 QAtesting/READER/testreader2.f
rw-r--r--1121/100 12183 Mar 18 15:25 1997 QAtesting/READER/testreader1.out
rw-r--r--1121/100 5146 Feb 25 08:15 1997 QAtesting/READER/testreader3.f
rw-r--r--1121/100 3652 Mar 18 15:35 1997 QAtesting/READER/inpfile4.inp
rw-r--r--1121/100 5458 Feb 25 08:52 1997 QAtesting/READER/testreader4.f
rw-r--r--1121/100 1070 Feb 6 10:59 1997 QAtesting/READER/READER_README
rw-r--r--1121/100 3665 Mar 8 15:27 1997 QAtesting/READER/TESTREAD
rw-r--r--1121/100 1666 Mar 18 15:23 1997 QAtesting/READER/inpfile1.inp
rw-r--r--1121/100 1361 Mar 18 15:31 1997 QAtesting/READER/inpfile2.inp
rw-r--r--1121/100 5185 Feb 25 09:23 1997 QAtesting/READER/testreader5.f
rw-r--r--1121/100 246 Feb 25 08:15 1997 QAtesting/READER/inpfile3.inp
rw-r--r--1121/100 6568 Mar 18 15:41 1997 QAtesting/READER/testreader2.out
rw-r--r--1121/100 3436 Mar 18 15:43 1997 QAtesting/READER/testreader3.out
rw-r--r--1121/100 19919 Mar 19 11:07 1997 QAtesting/READER/tpa.inp
rw-r--r--1121/100 49132 Mar 19 09:18 1997 QAtesting/READER/testreader4.out
rw-r--r--1121/100 10895 Mar 19 14:53 1997 QAtesting/READER/TESTREADER_README
rw-r--r--1121/100 161728 Mar 19 11:08 1997 QAtesting/READER/testreader5.out
rw-r--r--1121/100 3228 Mar 19 09:21 1997 QAtesting/READER/testreader6.out
rw-r--r--1121/100 8721 Mar 19 09:24 1997 QAtesting/READER/testreader7.out
rw-r--r--1121/100 3235 Mar 19 09:26 1997 QAtesting/READER/testreader8.out
rw-r--r--1121/100 2728 Mar 19 09:27 1997 QAtesting/READER/testreader9.out
rw-r--r--1121/100 6897 Mar 19 09:56 1997 QAtesting/READER/testreader10.out
rw-r--r--1121/100 5148 Feb 25 09:59 1997 QAtesting/READER/testreader8.f
rw-r--r--1121/100 5148 Feb 25 10:29 1997 QAtesting/READER/testreader14.f
rw-r--r--1121/100 219 Mar 18 15:37 1997 QAtesting/READER/inpfile8.inp
rw-r--r--1121/100 5330 Mar 4 07:23 1997 QAtesting/READER/testreader16.f
rw-r--r--1121/100 5147 Feb 25 09:31 1997 QAtesting/READER/testreader6.f
rw-r--r--1121/100 473 Mar 18 15:28 1997 QAtesting/READER/inpfile14.inp

| | | | | | | |
|-------------------|-------|-----|----|-------|------|-----------------------------------|
| rw-r--r--1121/100 | 172 | Mar | 18 | 15:36 | 1997 | Testing/READER/inpfile6.inp |
| rw-r--r--1121/100 | 408 | Mar | 18 | 15:29 | 1997 | QAtesting/READER/inpfile15.inp |
| rw-r--r--1121/100 | 511 | Mar | 18 | 15:37 | 1997 | QAtesting/READER/inpfile7.inp |
| rw-r--r--1121/100 | 5182 | Feb | 25 | 09:41 | 1997 | QAtesting/READER/testreader7.f |
| rw-r--r--1121/100 | 5209 | Feb | 25 | 10:54 | 1997 | QAtesting/READER/testreader15.f |
| rw-r--r--1121/100 | 506 | Mar | 18 | 15:37 | 1997 | QAtesting/READER/inpfile9.inp |
| rw-r--r--1121/100 | 5190 | Feb | 25 | 10:07 | 1997 | QAtesting/READER/testreader9.f |
| rw-r--r--1121/100 | 5189 | Feb | 25 | 10:17 | 1997 | QAtesting/READER/testreader10.f |
| rw-r--r--1121/100 | 740 | Mar | 18 | 15:27 | 1997 | QAtesting/READER/inpfile10.inp |
| rw-r--r--1121/100 | 3658 | Mar | 19 | 09:58 | 1997 | QAtesting/READER/testreader11.out |
| rw-r--r--1121/100 | 392 | Mar | 18 | 15:28 | 1997 | QAtesting/READER/inpfile11.inp |
| rw-r--r--1121/100 | 5218 | Feb | 25 | 10:46 | 1997 | QAtesting/READER/testreader11.f |
| rw-r--r--1121/100 | 10050 | Mar | 19 | 10:00 | 1997 | QAtesting/READER/testreader12.out |
| rw-r--r--1121/100 | 5218 | Feb | 25 | 11:04 | 1997 | QAtesting/READER/testreader12.f |
| rw-r--r--1121/100 | 411 | Mar | 18 | 15:28 | 1997 | QAtesting/READER/inpfile12.inp |
| rw-r--r--1121/100 | 5210 | Mar | 19 | 10:30 | 1997 | QAtesting/READER/testreader19.f |
| rw-r--r--1121/100 | 1045 | Mar | 18 | 15:28 | 1997 | QAtesting/READER/inpfile13.inp |
| rw-r--r--1121/100 | 5259 | Mar | 4 | 08:18 | 1997 | QAtesting/READER/testreader13.f |
| rw-r--r--1121/100 | 994 | Mar | 19 | 10:30 | 1997 | QAtesting/READER/inpfile19.inp |
| rw-r--r--1121/100 | 6864 | Mar | 19 | 10:05 | 1997 | QAtesting/READER/testreader13.out |
| rw-r--r--1121/100 | 5349 | Mar | 6 | 07:29 | 1997 | QAtesting/READER/testreader20.f |
| rw-r--r--1121/100 | 385 | Mar | 18 | 15:31 | 1997 | QAtesting/READER/inpfile20.inp |
| rw-r--r--1121/100 | 517 | Mar | 18 | 15:29 | 1997 | QAtesting/READER/inpfile16.inp |
| rw-r--r--1121/100 | 5349 | Mar | 4 | 07:35 | 1997 | QAtesting/READER/testreader17.f |
| rw-r--r--1121/100 | 388 | Mar | 18 | 15:29 | 1997 | QAtesting/READER/inpfile17.inp |
| rw-r--r--1121/100 | 5346 | Mar | 6 | 07:33 | 1997 | QAtesting/READER/testreader18.f |
| rw-r--r--1121/100 | 437 | Mar | 18 | 15:29 | 1997 | QAtesting/READER/inpfile18.inp |
| rw-r--r--1121/100 | 5208 | Mar | 6 | 07:42 | 1997 | QAtesting/READER/testreader21.f |
| rw-r--r--1121/100 | 316 | Mar | 18 | 15:31 | 1997 | QAtesting/READER/inpfile21.inp |
| rw-r--r--1121/100 | 5194 | Mar | 6 | 07:50 | 1997 | QAtesting/READER/testreader22.f |
| rw-r--r--1121/100 | 306 | Mar | 18 | 15:32 | 1997 | QAtesting/READER/inpfile22.inp |
| rw-r--r--1121/100 | 5288 | Mar | 6 | 07:57 | 1997 | QAtesting/READER/testreader23.f |
| rw-r--r--1121/100 | 358 | Mar | 18 | 15:32 | 1997 | QAtesting/READER/inpfile23.inp |
| rw-r--r--1121/100 | 5291 | Mar | 6 | 08:00 | 1997 | QAtesting/READER/testreader24.f |
| rw-r--r--1121/100 | 361 | Mar | 18 | 15:32 | 1997 | QAtesting/READER/inpfile24.inp |
| rw-r--r--1121/100 | 5224 | Mar | 6 | 08:11 | 1997 | QAtesting/READER/testreader25.f |
| rw-r--r--1121/100 | 319 | Mar | 18 | 15:32 | 1997 | QAtesting/READER/inpfile25.inp |
| rw-r--r--1121/100 | 4551 | Mar | 19 | 10:06 | 1997 | QAtesting/READER/testreader14.out |
| rw-r--r--1121/100 | 5224 | Mar | 6 | 08:17 | 1997 | QAtesting/READER/testreader26.f |
| rw-r--r--1121/100 | 401 | Mar | 18 | 15:32 | 1997 | QAtesting/READER/inpfile26.inp |
| rw-r--r--1121/100 | 5705 | Mar | 19 | 10:11 | 1997 | QAtesting/READER/testreader15.out |
| rw-r--r--1121/100 | 5237 | Mar | 6 | 08:21 | 1997 | QAtesting/READER/testreader27.f |
| rw-r--r--1121/100 | 322 | Mar | 18 | 15:33 | 1997 | QAtesting/READER/inpfile27.inp |
| rw-r--r--1121/100 | 3573 | Mar | 19 | 10:12 | 1997 | QAtesting/READER/testreader16.out |
| rw-r--r--1121/100 | 5226 | Mar | 6 | 08:46 | 1997 | QAtesting/READER/testreader28.f |
| rw-r--r--1121/100 | 341 | Mar | 18 | 15:33 | 1997 | QAtesting/READER/inpfile28.inp |
| rw-r--r--1121/100 | 3266 | Mar | 19 | 10:13 | 1997 | QAtesting/READER/testreader17.out |
| rw-r--r--1121/100 | 5224 | Mar | 6 | 09:00 | 1997 | QAtesting/READER/testreader29.f |
| rw-r--r--1121/100 | 327 | Mar | 18 | 15:33 | 1997 | QAtesting/READER/inpfile29.inp |
| rw-r--r--1121/100 | 4667 | Mar | 19 | 10:15 | 1997 | QAtesting/READER/testreader18.out |
| rw-r--r--1121/100 | 5232 | Mar | 6 | 09:05 | 1997 | QAtesting/READER/testreader30.f |
| rw-r--r--1121/100 | 330 | Mar | 18 | 15:33 | 1997 | QAtesting/READER/inpfile30.inp |
| rw-r--r--1121/100 | 6413 | Mar | 19 | 10:31 | 1997 | QAtesting/READER/testreader19.out |
| rw-r--r--1121/100 | 5232 | Mar | 6 | 09:08 | 1997 | QAtesting/READER/testreader31.f |
| rw-r--r--1121/100 | 330 | Mar | 18 | 15:33 | 1997 | QAtesting/READER/inpfile31.inp |
| rw-r--r--1121/100 | 3275 | Mar | 19 | 10:32 | 1997 | QAtesting/READER/testreader20.out |
| rw-r--r--1121/100 | 5228 | Mar | 6 | 09:10 | 1997 | QAtesting/READER/testreader32.f |
| rw-r--r--1121/100 | 298 | Mar | 18 | 15:33 | 1997 | QAtesting/READER/inpfile32.inp |
| rw-r--r--1121/100 | 5314 | Mar | 19 | 11:17 | 1997 | QAtesting/READER/testreader45.f |
| rw-r--r--1121/100 | 5228 | Mar | 6 | 09:14 | 1997 | QAtesting/READER/testreader33.f |

```

rw-r--r--1121/100 316 Mar 18 15:34 1997 QAtesting/READER/inpfile33.inp
rw-r--r--1121/100 286 Mar 19 11:18 1997 QAtesting/READER/inpfile45.inp
rw-r--r--1121/100 5234 Mar 6 09:54 1997 QAtesting/READER/testreader34.f
rw-r--r--1121/100 331 Mar 18 15:34 1997 QAtesting/READER/inpfile34.inp
rw-r--r--1121/100 2716 Mar 19 11:18 1997 QAtesting/READER/testreader45.out
rw-r--r--1121/100 5190 Mar 19 13:49 1997 QAtesting/READER/testreader35.f
rw-r--r--1121/100 1715 Mar 19 13:48 1997 QAtesting/READER/inpfile35.inp
rw-r--r--1121/100 4011 Mar 19 13:00 1997 QAtesting/READER/testreader21.out
rw-r--r--1121/100 5198 Mar 19 13:51 1997 QAtesting/READER/testreader36.f
rw-r--r--1121/100 1719 Mar 19 13:50 1997 QAtesting/READER/inpfile36.inp
rw-r--r--1121/100 4324 Mar 19 13:01 1997 QAtesting/READER/testreader22.out
rw-r--r--1121/100 5229 Mar 19 13:54 1997 QAtesting/READER/testreader37.f
rw-r--r--1121/100 1731 Mar 19 13:53 1997 QAtesting/READER/inpfile37.inp
rw-r--r--1121/100 4968 Mar 19 13:06 1997 QAtesting/READER/testreader23.out
rw-r--r--1121/100 5294 Mar 18 13:49 1997 QAtesting/READER/testreader38.f
rw-r--r--1121/100 1775 Mar 18 13:48 1997 QAtesting/READER/inpfile38.inp
rw-r--r--1121/100 4938 Mar 19 13:08 1997 QAtesting/READER/testreader24.out
rw-r--r--1121/100 5258 Mar 6 18:51 1997 QAtesting/READER/testreader39.f
rw-r--r--1121/100 385 Mar 18 15:35 1997 QAtesting/READER/inpfile39.inp
rw-r--r--1121/100 4839 Mar 19 13:10 1997 QAtesting/READER/testreader25.out
rw-r--r--1121/100 5257 Mar 6 18:56 1997 QAtesting/READER/testreader40.f
rw-r--r--1121/100 386 Mar 18 15:35 1997 QAtesting/READER/inpfile40.inp
rw-r--r--1121/100 5928 Mar 19 13:12 1997 QAtesting/READER/testreader26.out
rw-r--r--1121/100 5318 Mar 6 19:07 1997 QAtesting/READER/testreader41.f
rw-r--r--1121/100 371 Mar 18 15:36 1997 QAtesting/READER/inpfile41.inp
rw-r--r--1121/100 5864 Mar 19 13:14 1997 QAtesting/READER/testreader27.out
rw-r--r--1121/100 754 Mar 19 14:55 1997 QAtesting/READER/inpfile42.inp
rw-r--r--1121/100 5265 Mar 6 19:18 1997 QAtesting/READER/testreader42.f
rw-r--r--1121/100 5294 Mar 19 13:21 1997 QAtesting/READER/testreader28.out
rw-r--r--1121/100 3579 Mar 8 16:46 1997 QAtesting/READER/READERDOC_README
rw-r--r--1121/100 4898 Mar 19 13:22 1997 QAtesting/READER/testreader29.out
rw-r--r--1121/100 3492 Mar 19 13:23 1997 QAtesting/READER/testreader30.out
rw-r--r--1121/100 3307 Mar 19 13:42 1997 QAtesting/READER/testreader31.out
rw-r--r--1121/100 3360 Mar 19 13:43 1997 QAtesting/READER/testreader32.out
rw-r--r--1121/100 5189 Mar 7 09:23 1997 QAtesting/READER/testreader43.f
rw-r--r--1121/100 148 Mar 18 15:36 1997 QAtesting/READER/inpfile43.inp
rw-r--r--1121/100 4889 Mar 19 13:44 1997 QAtesting/READER/testreader33.out
rw-r--r--1121/100 5192 Mar 7 09:22 1997 QAtesting/READER/testreader44.f
rw-r--r--1121/100 230 Mar 18 15:36 1997 QAtesting/READER/inpfile44.inp
rw-r--r--1121/100 3247 Mar 19 13:46 1997 QAtesting/READER/testreader34.out
rw-r--r--1121/100 12672 Mar 19 13:50 1997 QAtesting/READER/testreader35.out
rw-r--r--1121/100 13316 Mar 19 13:52 1997 QAtesting/READER/testreader36.out
rw-r--r--1121/100 7895 Mar 19 13:54 1997 QAtesting/READER/testreader37.out
rw-r--r--1121/100 5400 Mar 19 13:57 1997 QAtesting/READER/testreader39.out
rw-r--r--1121/100 6109 Mar 19 13:58 1997 QAtesting/READER/testreader40.out
rw-r--r--1121/100 4927 Mar 19 14:13 1997 QAtesting/READER/testreader41.out
rw-r--r--1121/100 5568 Mar 19 14:56 1997 QAtesting/READER/testreader42.out
rw-r--r--1121/100 3303 Mar 19 14:42 1997 QAtesting/READER/testreader43.out
rw-r--r--1121/100 4886 Mar 19 14:43 1997 QAtesting/READER/testreader44.out
rwxr-xr-x1121/100 0 Mar 20 11:33 1997 QAtesting/ebsfail/
rw-r--r--1121/100 2126 Mar 20 11:33 1997 QAtesting/ebsfail/ebsfailtesting.readme
rw-r--r--1121/100 2306 Mar 20 11:33 1997 QAtesting/ebsfail/test1.readme
rw-r--r--1121/100 5047 Mar 20 11:33 1997 QAtesting/ebsfail/test1chloride.dat
rw-r--r--1121/100 9962 Mar 20 11:33 1997 QAtesting/ebsfail/test1corrode.out
rw-r--r--1121/100 4636 Mar 20 11:33 1997 QAtesting/ebsfail/test1ebsfail.inp
rw-r--r--1121/100 9923 Mar 20 11:33 1997 QAtesting/ebsfail/test1ebstrhc.inp
rw-r--r--1121/100 9962 Mar 20 11:33 1997 QAtesting/ebsfail/test1failtcorrode.out
rw-r--r--1121/100 10968 Mar 20 11:33 1997 QAtesting/ebsfail/test1failttemphumd.dat
rw-r--r--1121/100 10968 Mar 20 11:33 1997 QAtesting/ebsfail/test1temphumd.dat
rw-r--r--1121/100 53352 Mar 20 11:33 1997 QAtesting/ebsfail/test1tpa.inp

```

```

rw-r--r--1121/100 4479 Mar 20 11:33 1997 .esting/ebsfail/test2.readme
rw-r--r--1121/100 5047 Mar 20 11:33 1997 QAtesting/ebsfail/test2chloride.dat
rw-r--r--1121/100 170532 Mar 20 11:33 1997 QAtesting/ebsfail/test2multiflo.dat
rw-r--r--1121/100 1164 Mar 20 11:33 1997 QAtesting/ebsfail/test2readmultiflo.f
rw-r--r--1121/100 4467 Mar 20 11:33 1997 QAtesting/ebsfail/test3.readme
rw-r--r--1121/100 4636 Mar 20 11:33 1997 QAtesting/ebsfail/test3ebsfail.inp
rw-r--r--1121/100 4168 Mar 20 11:33 1997 QAtesting/ebsfail/test3ebsfail.tmp
rw-r--r--1121/100 3776 Mar 20 11:33 1997 QAtesting/ebsfail/test4.f
rw-r--r--1121/100 10968 Mar 20 11:33 1997 QAtesting/ebsfail/test3temphumd.dat
rw-r--r--1121/100 53323 Mar 20 11:33 1997 QAtesting/ebsfail/test3tpa.inp
rw-r--r--1121/100 5785 Mar 20 11:33 1997 QAtesting/ebsfail/test4.inp
rw-r--r--1121/100 85560 Mar 20 11:33 1997 QAtesting/ebsfail/test4.out
rw-r--r--1121/100 2561 Mar 20 11:33 1997 QAtesting/ebsfail/test4.readme
rw-r--r--1121/100 5047 Mar 20 11:33 1997 QAtesting/ebsfail/test4chloride.dat
rw-r--r--1121/100 14506 Mar 20 11:33 1997 QAtesting/ebsfail/test4corrode.out
rw-r--r--1121/100 4636 Mar 20 11:33 1997 QAtesting/ebsfail/test4ebsfail.inp
rw-r--r--1121/100 9923 Mar 20 11:33 1997 QAtesting/ebsfail/test4ebstrhc.inp
rw-r--r--1121/100 71199 Mar 20 11:33 1997 QAtesting/ebsfail/test4failt.f
rwxr-xr-x1121/100 0 Mar 31 13:55 1997 QAtesting/exec/
rw-r--r--1121/100 27011 Mar 14 14:19 1997 QAtesting/exec/exec.f
rw-r--r--1121/100 8911 Mar 14 14:19 1997 QAtesting/exec/testexec.f
rw-r--r--1121/100 23573 Mar 14 14:19 1997 QAtesting/exec/testexec.out
rw-r--r--1121/100 2477 Mar 14 14:19 1997 QAtesting/exec/testexec_README
rw-r--r--1121/100 46618 Mar 14 14:20 1997 QAtesting/exec/tpa.inp
rwxr-xr-x1121/100 0 Mar 14 07:17 1997 QAtesting/dcagw/
rw-r--r--1121/100 5212 Mar 6 13:25 1997 QAtesting/dcagw/testdcagw30a.f
rw-r--r--1121/100 5082 Mar 6 12:57 1997 QAtesting/dcagw/testdcagw5a.f
rw-r--r--1121/100 2527 Mar 6 09:58 1997 QAtesting/dcagw/TESTDCAGW_README
rw-r--r--1121/100 2727 Mar 6 13:27 1997 QAtesting/dcagw/testdcagw30a.out
rw-r--r--1121/100 2540 Mar 6 13:29 1997 QAtesting/dcagw/testdcagw5a.out
rw-r--r--1121/100 5081 Mar 6 13:02 1997 QAtesting/dcagw/testdcagw5b.f
rw-r--r--1121/100 2540 Mar 6 13:29 1997 QAtesting/dcagw/testdcagw5b.out
rw-r--r--1121/100 3975 Mar 6 12:58 1997 QAtesting/dcagw/testdcagwerr.f
rw-r--r--1121/100 25172 Mar 6 13:37 1997 QAtesting/dcagw/testdcagwerr.out
rw-r--r--1121/100 2350 Mar 6 13:39 1997 QAtesting/dcagw/testdcagwerra.out
rw-r--r--1121/100 1923 Mar 6 13:40 1997 QAtesting/dcagw/testdcagwerrb.out
rwxr-xr-x1121/100 0 Mar 6 14:42 1997 QAtesting/ashrmovo/
rw-r--r--1121/100 91135 Mar 4 14:05 1997 QAtesting/ashrmovo/testarbigUKd.out
rw-r--r--1121/100 91246 Mar 6 14:56 1997 QAtesting/ashrmovo/testarbigashdensity.out
rw-r--r--1121/100 91237 Mar 6 15:03 1997 QAtesting/ashrmovo/testarbigblankrem.out
rw-r--r--1121/100 16797 Mar 6 14:42 1997 QAtesting/ashrmovo/ashremovetpa.f
rw-r--r--1121/100 91228 Mar 4 14:07 1997 QAtesting/ashrmovo/testarbigmoisturefract.out
rw-r--r--1121/100 91281 Mar 6 15:09 1997 QAtesting/ashrmovo/testarbigrain.out
rw-r--r--1121/100 91265 Mar 6 15:11 1997 QAtesting/ashrmovo/testarbigsoildepth.out
rw-r--r--1121/100 91351 Mar 4 14:07 1997 QAtesting/ashrmovo/testarbigtimeofevent.out
rw-r--r--1121/100 89405 Mar 4 13:44 1997 QAtesting/ashrmovo/testardef.out
rw-r--r--1121/100 91256 Mar 6 15:14 1997 QAtesting/ashrmovo/testarlowUso1.out
rw-r--r--1121/100 3180 Mar 6 15:35 1997 QAtesting/ashrmovo/testashremove_README
rw-r--r--1121/100 91278 Mar 6 15:06 1997 QAtesting/ashrmovo/testarbigfsat.out
rw-r--r--1121/100 89405 Mar 6 15:18 1997 QAtesting/ashrmovo/testarlowtime.out
rw-r--r--1121/100 2186 Mar 6 14:42 1997 QAtesting/ashrmovo/testashremove.f
rwxr-xr-x1121/100 0 Mar 8 11:44 1997 QAtesting/FILEUNIT/
rw-r--r--1121/100 1275 Mar 8 11:44 1997 QAtesting/FILEUNIT/testfileunit1.f
rw-r--r--1121/100 2284 Mar 8 11:44 1997 QAtesting/FILEUNIT/testfileunit1.out
rw-r--r--1121/100 3035 Mar 8 11:44 1997 QAtesting/FILEUNIT/testfileunit2.f
rw-r--r--1121/100 1130 Mar 8 11:44 1997 QAtesting/FILEUNIT/testfileunit2.out
rw-r--r--1121/100 3051 Mar 8 11:44 1997 QAtesting/FILEUNIT/testfileunit3.f
rw-r--r--1121/100 8321 Mar 8 11:44 1997 QAtesting/FILEUNIT/testfileunit3.out
rw-r--r--1121/100 5929 Mar 8 11:44 1997 QAtesting/FILEUNIT/testfileunit4.f
rw-r--r--1121/100 3092 Mar 8 11:44 1997 QAtesting/FILEUNIT/testfileunit4.out

```

| | | | | | | |
|-------------------|---------|-----|----|-------|------|--|
| rw-r--r--1121/100 | 2305 | Mar | 8 | 11:44 | 1997 | Testing/FILEUNIT/FILEUNITDOC_README |
| rw-r--r--1121/100 | 1389 | Mar | 8 | 11:44 | 1997 | QAtesting/FILEUNIT/TESTFILEUNIT_README |
| rw-r--r--1121/100 | 0 | Apr | 1 | 10:50 | 1997 | QAtesting/ebsrel/ |
| rw-r--r--1121/100 | 5913 | Mar | 31 | 17:30 | 1997 | QAtesting/ebsrel/ebsrel_001.inp |
| rw-r--r--1121/100 | 5913 | Mar | 31 | 17:30 | 1997 | QAtesting/ebsrel/ebsrel_002.inp |
| rw-r--r--1121/100 | 5913 | Mar | 31 | 17:30 | 1997 | QAtesting/ebsrel/ebsrel_003.inp |
| rw-r--r--1121/100 | 651 | Mar | 31 | 17:30 | 1997 | QAtesting/ebsrel/relfrac_001.out |
| rw-r--r--1121/100 | 651 | Mar | 31 | 17:30 | 1997 | QAtesting/ebsrel/relfrac_002.out |
| rw-r--r--1121/100 | 651 | Mar | 31 | 17:30 | 1997 | QAtesting/ebsrel/relfrac_003.out |
| rw-r--r--1121/100 | 53264 | Mar | 31 | 17:30 | 1997 | QAtesting/ebsrel/tpa_001.inp |
| rw-r--r--1121/100 | 53277 | Mar | 31 | 17:30 | 1997 | QAtesting/ebsrel/tpa_002.inp |
| rw-r--r--1121/100 | 53279 | Mar | 31 | 17:30 | 1997 | QAtesting/ebsrel/tpa_003.inp |
| rw-r--r--1121/100 | 1381 | Apr | 1 | 13:45 | 1997 | QAtesting/ebsrel/EBSREL_README |
| rw-r--r--1121/100 | 0 | Apr | 1 | 09:55 | 1997 | QAtesting/dcags/ |
| rw-r--r--1121/100 | 4525 | Feb | 28 | 11:25 | 1997 | QAtesting/dcags/testdcags.f |
| rw-r--r--1121/100 | 4090 | Feb | 28 | 11:23 | 1997 | QAtesting/dcags/testdcags.out |
| rw-r--r--1121/100 | 617 | Apr | 1 | 10:43 | 1997 | QAtesting/dcags/dcags_README |
| rw-r--r--1121/100 | 0 | Mar | 8 | 11:47 | 1997 | QAtesting/FINDELEV/ |
| rw-r--r--1121/100 | 4266 | Mar | 8 | 11:47 | 1997 | QAtesting/FINDELEV/testfindelev.f |
| rw-r--r--1121/100 | 40349 | Mar | 8 | 11:47 | 1997 | QAtesting/FINDELEV/testfindelev.out |
| rw-r--r--1121/100 | 1180 | Mar | 8 | 11:47 | 1997 | QAtesting/FINDELEV/FINDELEVDOC_README |
| rw-r--r--1121/100 | 1170 | Mar | 8 | 11:47 | 1997 | QAtesting/FINDELEV/TESTFINDELEV_README |
| rw-r--r--1121/100 | 0 | Mar | 8 | 14:29 | 1997 | QAtesting/MV/ |
| rw-r--r--1121/100 | 10025 | Mar | 5 | 18:45 | 1997 | QAtesting/MV/testmv.f |
| rw-r--r--1121/100 | 6732 | Jan | 29 | 14:32 | 1997 | QAtesting/MV/testmv.out |
| rw-r--r--1121/100 | 2173 | Feb | 7 | 10:23 | 1997 | QAtesting/MV/MV_README |
| rw-r--r--1121/100 | 4163 | Mar | 8 | 14:24 | 1997 | QAtesting/MV/TESTMV_README |
| rw-r--r--1121/100 | 6625 | Mar | 8 | 14:29 | 1997 | QAtesting/MV/MVDOC_README |
| rw-r--r--1121/100 | 0 | Apr | 1 | 10:38 | 1997 | QAtesting/seismo/ |
| rw-r--r--1121/100 | 2413 | Mar | 31 | 14:27 | 1997 | QAtesting/seismo/testseismo.f |
| rw-r--r--1121/100 | 6044 | Mar | 31 | 14:28 | 1997 | QAtesting/seismo/seismo.f |
| rw-r--r--1121/100 | 6669 | Mar | 31 | 14:29 | 1997 | QAtesting/seismo/failure.f |
| rw-r--r--1121/100 | 5896 | Mar | 31 | 14:29 | 1997 | QAtesting/seismo/impact.f |
| rw-r--r--1121/100 | 5904 | Mar | 31 | 14:30 | 1997 | QAtesting/seismo/stress.f |
| rw-r--r--1121/100 | 85088 | Apr | 1 | 09:15 | 1997 | QAtesting/seismo/seismo.wp |
| rw-r--r--1121/100 | 1252 | Apr | 1 | 10:38 | 1997 | QAtesting/seismo/seismo_README |
| rw-r--r--1121/100 | 0 | Mar | 21 | 10:43 | 1997 | QAtesting/volcano/ |
| rw-r--r--1121/100 | 1053 | Mar | 20 | 17:10 | 1997 | QAtesting/volcano/testvolcano1.inp |
| rw-r--r--1121/100 | 6411 | Mar | 20 | 17:14 | 1997 | QAtesting/volcano/testvolcano1.f |
| rw-r--r--1121/100 | 2485 | Mar | 21 | 11:01 | 1997 | QAtesting/volcano/TESTVOLCANO_README |
| rw-r--r--1121/100 | 6647 | Mar | 20 | 17:14 | 1997 | QAtesting/volcano/testvolcano2.f |
| rw-r--r--1121/100 | 11380 | Mar | 20 | 17:14 | 1997 | QAtesting/volcano/testvolcano1.out |
| rw-r--r--1121/100 | 1054 | Mar | 20 | 17:15 | 1997 | QAtesting/volcano/testvolcano2.inp |
| rw-r--r--1121/100 | 11546 | Mar | 20 | 17:16 | 1997 | QAtesting/volcano/testvolcano2.out |
| rw-r--r--1121/100 | 7469 | Mar | 21 | 08:16 | 1997 | QAtesting/volcano/testvolcano3.f |
| rw-r--r--1121/100 | 1258 | Mar | 21 | 08:10 | 1997 | QAtesting/volcano/testvolcano3.inp |
| rw-r--r--1121/100 | 18209 | Mar | 21 | 08:21 | 1997 | QAtesting/volcano/testvolcano3.out |
| rw-r--r--1121/100 | 7384 | Mar | 21 | 09:01 | 1997 | QAtesting/volcano/testvolcano4.f |
| rw-r--r--1121/100 | 1258 | Mar | 21 | 09:03 | 1997 | QAtesting/volcano/testvolcano4.inp |
| rw-r--r--1121/100 | 18097 | Mar | 21 | 09:04 | 1997 | QAtesting/volcano/testvolcano4.out |
| rw-r--r--1121/100 | 7620 | Mar | 21 | 09:28 | 1997 | QAtesting/volcano/testvolcano5.f |
| rw-r--r--1121/100 | 1260 | Mar | 21 | 10:04 | 1997 | QAtesting/volcano/testvolcano5.inp |
| rw-r--r--1121/100 | 18311 | Mar | 21 | 10:05 | 1997 | QAtesting/volcano/testvolcano5.out |
| rw-r--r--1121/100 | 0 | Mar | 14 | 11:21 | 1997 | QAtesting/nfenv/ |
| rw-r--r--1121/100 | 1274984 | Mar | 14 | 10:41 | 1997 | QAtesting/nfenv/multiflo.dat |
| rw-r--r--1121/100 | 22085 | Mar | 14 | 10:33 | 1997 | QAtesting/nfenv/nfenv.f |
| rw-r--r--1121/100 | 14169 | Mar | 14 | 10:40 | 1997 | QAtesting/nfenv/testnfenv.f |
| rw-r--r--1121/100 | 80936 | Mar | 14 | 10:39 | 1997 | QAtesting/nfenv/testnfenv.o |
| rw-r--r--1121/100 | 7601 | Mar | 14 | 10:41 | 1997 | QAtesting/nfenv/testnfenv.out |
| rw-r--r--1121/100 | 1628 | Mar | 14 | 13:24 | 1997 | QAtesting/nfenv/testnfenv_README |

```

rw-r--r--1121/100 45812 Mar 13 16:58 1997 QAtesting/nfenv/tpa.inp
rw-r--r--1121/100 61 Mar 13 16:57 1997 QAtesting/nfenv/maxchain.i
rw-r--r--1121/100 135 Mar 13 16:57 1997 QAtesting/nfenv/maxnnucl.i
rw-r--r--1121/100 37 Mar 13 16:57 1997 QAtesting/nfenv/maxnsuba.i
rw-r--r--1121/100 35 Mar 13 16:57 1997 QAtesting/nfenv/maxntime.i
rw-r--r--1121/100 934904 Mar 13 16:57 1997 QAtesting/nfenv/multiflo.out
rw-r--r--1121/100 52496 Mar 14 10:39 1997 QAtesting/nfenv/nfenv.o
rw-r--r--1121/100 14004 Mar 14 09:23 1997 QAtesting/nfenv/testnfenvB.out
rwxr-xr-x1121/100 0 Mar 31 13:45 1997 QAtesting/uzft_szft/
rw-r--r--1121/100 5894 Mar 31 13:45 1997 QAtesting/uzft_szft/README
rw-r--r--1121/100 96 Mar 31 13:45 1997 QAtesting/uzft_szft/README_FIRST
rw-r--r--1121/100 430 Mar 31 13:45 1997 QAtesting/uzft_szft/Snakefile
rw-r--r--1121/100 231564 Mar 31 13:45 1997 QAtesting/uzft_szft/archive.a
rw-r--r--1121/100 368640 Mar 31 13:45 1997 QAtesting/uzft_szft/eval.x
rw-r--r--1121/100 4882 Mar 31 13:45 1997 QAtesting/uzft_szft/evaluate.f
rw-r--r--1121/100 1934 Mar 31 13:45 1997 QAtesting/uzft_szft/function.f
rw-r--r--1121/100 14915 Mar 31 13:45 1997 QAtesting/uzft_szft/nefiisz.inp
rw-r--r--1121/100 46736 Mar 31 13:45 1997 QAtesting/uzft_szft/nefiisz.out
rw-r--r--1121/100 136096 Mar 31 13:45 1997 QAtesting/uzft_szft/nefiisz.src
rw-r--r--1121/100 8908 Mar 31 13:45 1997 QAtesting/uzft_szft/nefiuiz.inp
rw-r--r--1121/100 2158624 Mar 31 13:45 1997 QAtesting/uzft_szft/nefiuiz.out
rw-r--r--1121/100 19540 Mar 31 13:45 1997 QAtesting/uzft_szft/nefiuiz.out_short_version
rw-r--r--1121/100 139009 Mar 31 13:45 1997 QAtesting/uzft_szft/nefiuiz.src
rw-r--r--1121/100 249 Mar 31 13:45 1997 QAtesting/uzft_szft/nefiuiz.vel
rw-r--r--1121/100 2347 Mar 31 13:45 1997 QAtesting/uzft_szft/props.f
rw-r--r--1121/100 42972 Mar 31 13:45 1997 QAtesting/uzft_szft/sp.tpa
rw-r--r--1121/100 186 Mar 31 13:45 1997 QAtesting/uzft_szft/test_tpa.dat
rw-r--r--1121/100 14241 Mar 31 13:45 1997 QAtesting/uzft_szft/test_tpa.out
rw-r--r--1121/100 45550 Mar 31 13:45 1997 QAtesting/uzft_szft/tpa.inp
rwxr-xr-x1121/100 0 Apr 1 11:33 1997 QAtesting/faulto/
rw-r--r--1121/100 2164 Mar 31 14:39 1997 QAtesting/faulto/testfaulto_README
rw-r--r--1121/100 2012 Mar 31 14:42 1997 QAtesting/faulto/testfaulto.f
rw-r--r--1121/100 7931 Mar 31 14:43 1997 QAtesting/faulto/faulto.f
rw-r--r--1121/100 28412 Apr 1 11:33 1997 QAtesting/faulto/summary.wp5
rw-r--r--1121/100 27430 Mar 31 14:46 1997 QAtesting/faulto/faulto1_1.out
rw-r--r--1121/100 27036 Mar 31 14:46 1997 QAtesting/faulto/faulto1_2.out
rw-r--r--1121/100 17085 Mar 31 14:46 1997 QAtesting/faulto/faulto2_1.out
rw-r--r--1121/100 17608 Mar 31 14:46 1997 QAtesting/faulto/faulto2_2.out
rw-r--r--1121/100 16992 Mar 31 14:46 1997 QAtesting/faulto/faulto2_3.out
rw-r--r--1121/100 16748 Mar 31 14:46 1997 QAtesting/faulto/faulto2_4.out
rw-r--r--1121/100 16926 Mar 31 14:46 1997 QAtesting/faulto/faulto2_5.out
rw-r--r--1121/100 9517 Mar 31 14:46 1997 QAtesting/faulto/faulto3_1.out
rw-r--r--1121/100 8618 Mar 31 14:46 1997 QAtesting/faulto/faulto3_2.out
rw-r--r--1121/100 3534 Mar 31 14:46 1997 QAtesting/faulto/faulto4_1.out
rw-r--r--1121/100 3038 Mar 31 14:46 1997 QAtesting/faulto/faulto4_2.out
rwxr-xr-x1121/100 0 Mar 31 17:18 1997 QAtesting/uzflow_climato/
rw-rw-rw-1121/100 78384 Mar 31 17:18 1997 QAtesting/uzflow_climato/tempnote.dvi
rwxr-xr-x1121/100 0 Mar 6 14:32 1997 QAtesting/ashplumo/
rw-r--r--1121/100 2937 Mar 3 16:49 1997 QAtesting/ashplumo/testashplume_README
rw-r--r--1121/100 22293 Mar 3 16:50 1997 QAtesting/ashplumo/testashplume.out
rw-r--r--1121/100 757 Mar 6 14:30 1997 QAtesting/ashplumo/testashplume.f
rw-r--r--1121/100 4337 Mar 6 14:32 1997 QAtesting/ashplumo/ashplumetpa.f
rw-r--r--1121/100 51569 Apr 1 16:26 1997 QAtesting/_QA_TESTING_README

```

**SOFTWARE
REQUIREMENTS
DESCRIPTION**

6/7/95

**TOTAL PERFORMANCE ASSESSMENT (TPA) CODE
VERSION 3.0 SOFTWARE REQUIREMENTS DESCRIPTION**

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

Prepared by

Randall Manteufel

RG Baeza
EM for PA

TOTAL PERFORMANCE ASSESSMENT (TPA) CODE VERSION 3.0 SOFTWARE REQUIREMENTS DESCRIPTION

1 INTRODUCTION

This software requirements description is the first step in updating the Total Performance Assessment (TPA) code from version 2.0 to 3.0. The TPA code Version 2.0 (Sagar and Janetzke, 1993) was used in the Nuclear Regulatory Commission/Center for Nuclear Waste Regulatory Analyses (NRC/CNWRA) Iterative Performance Assessment (IPA) Phase 2 exercise. The TPA code is an executive module and a set of consequence modules that simulate the performance of a geologic repository of nuclear high-level waste (HLW) at Yucca Mountain (YM), Nevada. The executive module controls the flow of data and execution between the process/component-specific consequence modules that simulate major safety components of the repository system. The TPA code integrates geologic site characterization data, proposed repository and waste package (WP) engineered designs, and biosphere data. The consequence modules are designed with input from materials engineers, hydrogeologists, seismologists, volcanologists, rock mechanicians, and health physicists. Recent developments in the proposed YM repository necessitate a new version of TPA code. These developments include (i) new repository, WP, and emplacement designs, (ii) changing regulatory standards (from release-based to dose-based), and (iii) potentially longer time periods of concern (to hundreds of thousands of years). Numerous improvements will be incorporated that reflect knowledge and data gained in recent years of site characterization and laboratory studies, as well as other total system performance assessments (TSPA). The TPA code will have the capability to assess the compliance of the proposed YM repository with regulatory requirements using a probabilistic approach to account for uncertainties.

2 SOFTWARE FUNCTION

The TPA code is a combination of an executive module and a set of consequence modules that stochastically assess the overall performance of the proposed YM HLW repository with applicable regulatory standards. The executive driver controls the probabilistic sampling of input parameters, the calculational flow process between modules, and the generation of output files. Output files can be used for parameter importance analyses, generation of time-dependent risk curves, and generation of complementary cumulative distribution functions for cumulative release of radionuclides. Utility modules ensure a consistent description of the proposed repository system and flow of data between consequence modules. Examples include the spatial and temporal discretizations (i.e., subarea (SA) discretization of the proposed repository and time stepping scheme). In the NRC/CNWRA IPA Phase 2 exercise, there were 7 SAs and 50 time steps of 200 yr each. The number of SAs in the TPA code will be based on the latest proposed repository design and reflect near-field thermal-hydrologic-mechanical-chemical (THMC) environments in the proposed repository as well as hydrostratigraphy. The time stepping will be variable over the simulation as well as the total time period of interest (TPI).

3 TECHNICAL AND COMPUTATIONAL APPROACHES

The overall conceptual approach of a TSPA is outlined in figure 1. Data flow from the system characterization to final regulatory compliance determination (shown as either a cumulative release or dose). The bulk of the modeling effort is in the consequence modules that include both anticipated

69/95

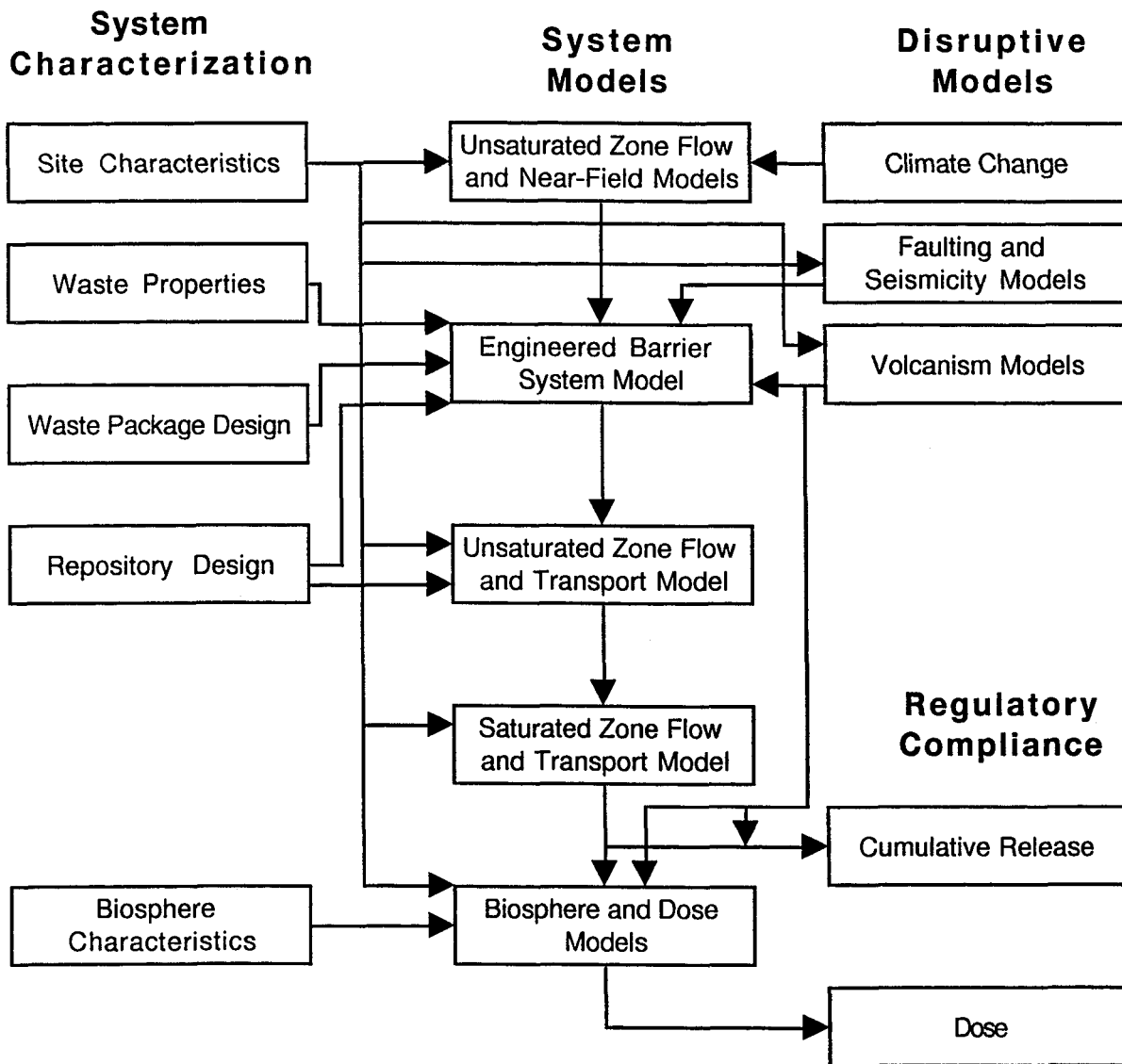


Figure 1. Overall TSPA flow diagram

processes (also called base-case processes) and disruptive processes. The base-case system has seven major subsystem models:

- groundwater flow from the ground surface to the proposed repository,
- near-field THMC environment of the engineered barrier system (EBS),
- corrosion and other anticipated failure mechanisms of the EBS containment,
- release of radionuclides from the EBS into geologic setting,
- groundwater flow and radionuclide transport in the unsaturated zone below the proposed repository and into the saturated zone,
- groundwater flow and radionuclide transport in the saturated zone below the proposed repository to a compliance point (CP) or boundary, and
- transport of radionuclides in the biosphere through the groundwater pathway that leads to dose to humans.

The disruptive system has faulting, seismicity, volcanism and climate change that cause earlier failures of EBS containment. In the case of volcanism, radionuclides may be released directly into the accessible environment and at the CP through the ground surface pathway.

A number of utility modules will be used to provide general data and generic capabilities that more than one module may need. For example, the initial radionuclide inventory will be calculated in a module so this information can be provided to other modules.

The TPA code will control the spatial discretization of the proposed repository (i.e., number of SAs), the distance from the proposed repository to the CP (e.g., 5, 20, 25, or 30 km), the temporal discretization scheme (e.g., output every 200 yr), and the TPI (e.g., 100,000 yr).

Major analysis improvements in the TPA code include the ability to

- increase or decrease the TPI
- evaluate finer time discretizations using nonuniform time steps
- evaluate finer repository spatial discretizations
- evaluate different areal mass loading
- calculate time-dependent dose rate at a CP
- calculate peak dose rate at a CP in the TPI
- evaluate dilution in saturated zone
- evaluate in-drift emplacement design
- add or remove sampled parameters

The TPA code Version 3.0 will also include:

- updated consequence models
- improved parameter importance analysis capabilities
- streamlined scope for consequence modules
- streamlined methodology for data transfer between consequence modules
- more flexible design to accommodate changes in consequence modules

7/1/95

4 USER INTERFACE AND DATA FLOW

The TPA code will be executed in file batch mode using one main input file: "tpa.inp." The TPA code reads data from this one input file only. The TPA code writes output data into a set of files for plotting or importance analyses. The file interfaces are described here.

4.1 INPUT TO TOTAL PERFORMANCE ASSESSMENT CODE

All of the input data for the TPA code is contained in the "tpa.inp" file. No other files/input will have an effect on the TPA executive (EXEC) calculations. Auxiliary files for data may be needed for consequence modules, however, these files should be "static" and the "tpa.inp" file should be used for parameter descriptions that change from run to run. The "tpa.inp" file contains all parameters necessary to describe the scenario and the number of realizations requested. The "tpa.inp" file starts with two comment lines that the analysts should use to describe the type of run being performed. These lines will be read as Character*80 and echoed at the top of all output files.

4.2 FLOW OF DATA BETWEEN MODULES

EXEC controls data flow by passing data in the subroutine call statement to each module. EXEC does not use common blocks or disk files for data transfer between itself and consequence modules. Within a consequence module, common blocks or files can be used. EXEC does not permit that data be passed directly between consequence modules. Each module is to be called only by EXEC and not by other modules. For efficiency and implementation purposes, the modules can consist of more than one subroutine, may call TPA code utility subroutines (e.g., INVENT), or may call stand alone programs (e.g., NEFTRAN). But modules are to pass information only to EXEC to control the simulation process.

The overall sequence of execution and flow of data is shown in figure 2. Here, EXEC starts the simulation by reading the "tpa.inp" file through the READER routine. The READER module need only be called once during a run. Having determined the parameters that describe the system, the EXEC continues by calling component specific consequence modules. Some modules will only be called once during one realization, while others will be called many times. Modules being called once include SAMPLER, SZFT, DCAGW and DCAGS. The modules UZFLOW, NFENV, EBSFAIL, EBSREL, and UZFT will be called once for each SA for each realization. If disruptive scenarios are being analyzed, the FAULTING, SEISMO, VOLCANO, ASHPLUME, and DCAGS modules will be called directly by EXEC once during a realization. These disruptive modules will be used to either cause earlier failure of the EBS or provide a more direct pathway for radionuclides into the biosphere (e.g., VOLCANO through ASHPLUME). If desired in the future, disruptive scenarios can be designed or modified to affect groundwater flow and transport of radionuclides.

The computational scheme for the TPA code is shown in figure 3. Each module is called by the EXEC module with some inputs and then returns some outputs to the EXEC through the call statement. Consequence modules do not call each other directly. There is a clear expectation of inputs and outputs between the EXEC and each module. In most cases, the output of one module is the input to the next module. The EXEC has two main loops for the number of realizations and the number of SAs.

The utility modules and consequence modules are discussed next.

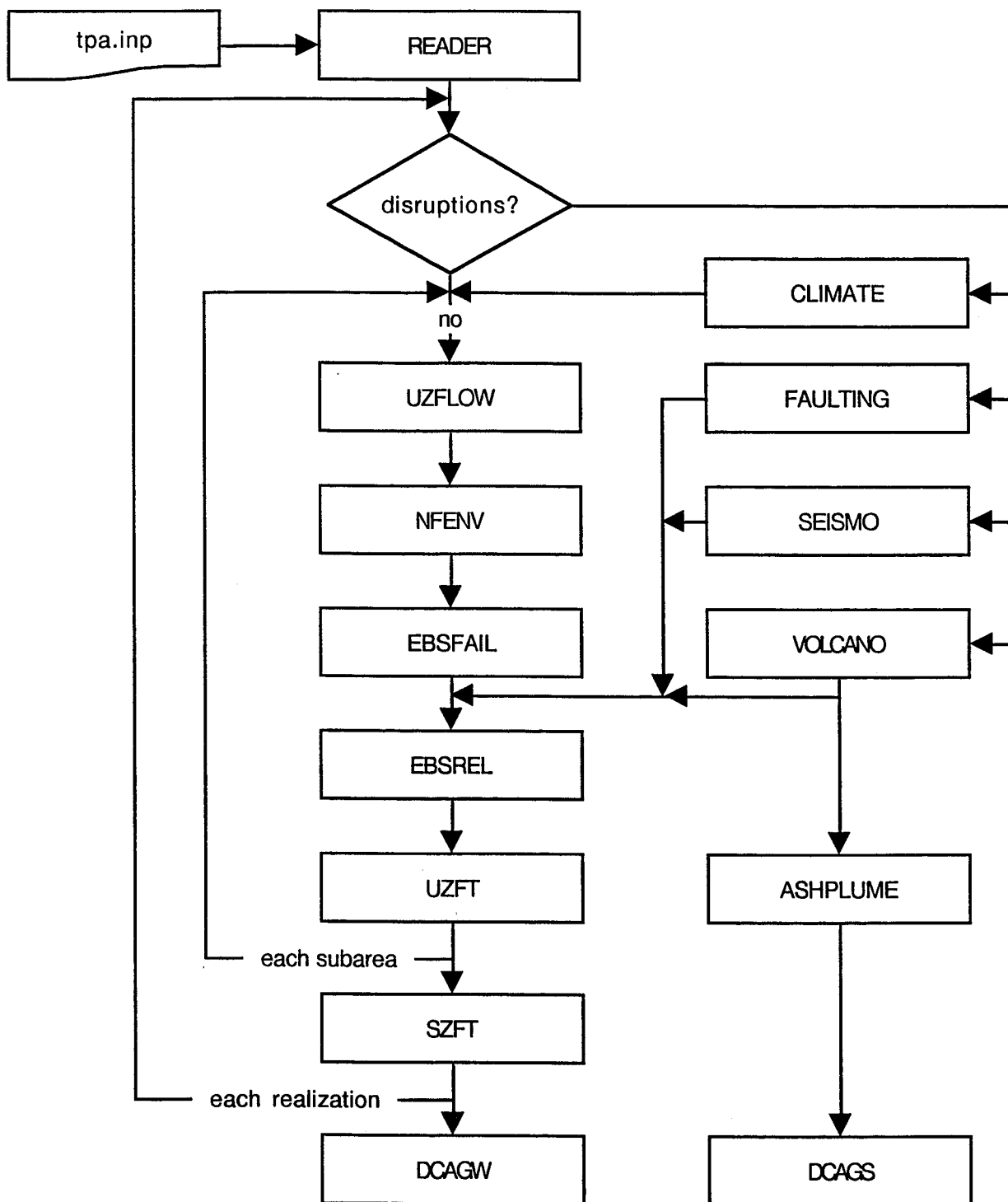


Figure 2. TPA flow diagram

73/95

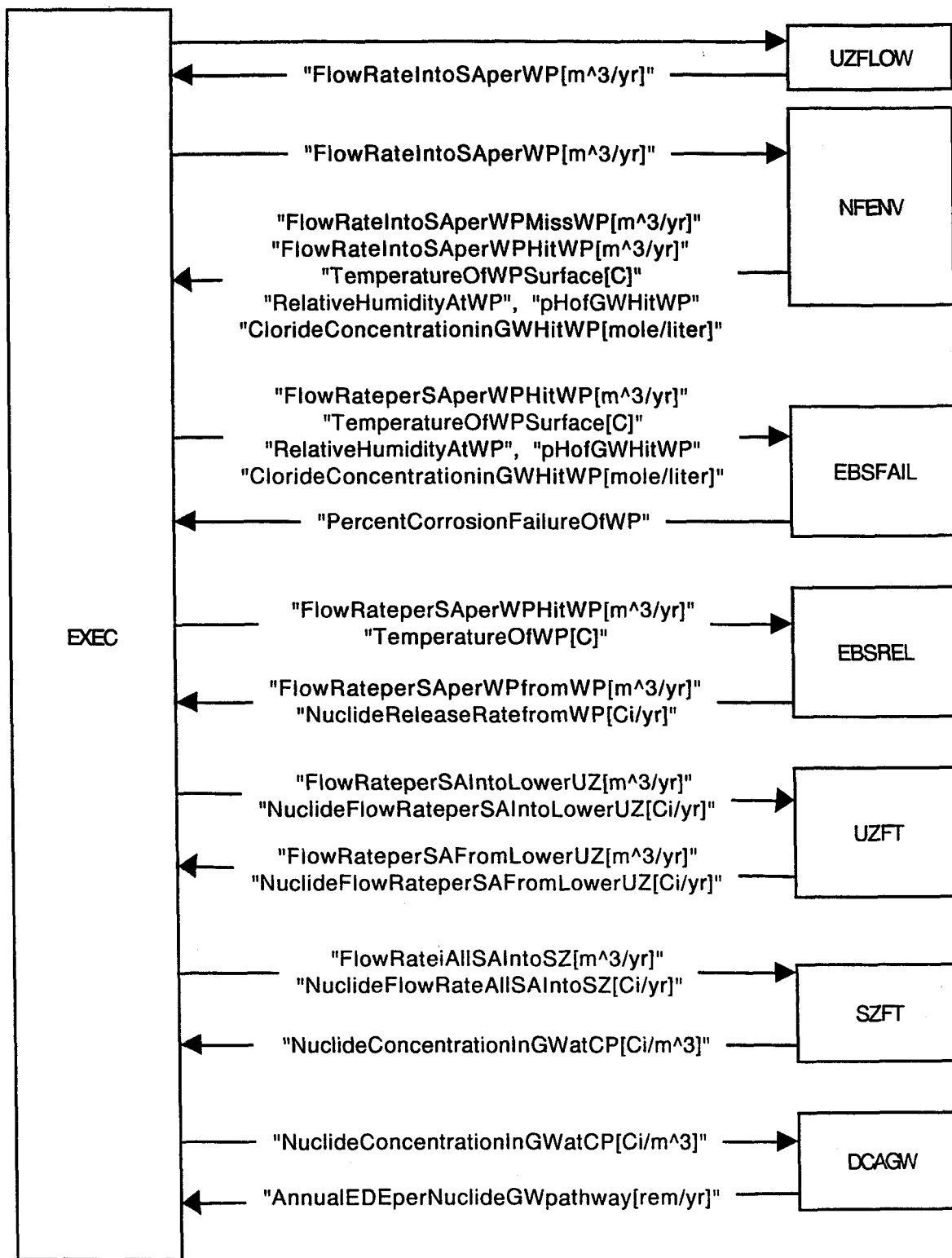


Figure 3. Main input/output associated with base-case flow and transport

4.3 UTILITY MODULES

4.3.1 Reader

READER is a utility module that preprocesses the data from the "tpa.inp" file. This is the only subroutine that reads the "tpa.inp" file. This module is similar to that already existing in the TPA code Version 2.0. The "tpa.inp" file will contain data specific for the TPA code execution as well as all probability distribution function (PDF) definitions for parameters that will be provided to the consequence modules.

4.3.2 Sampled Parameter

This module dynamically stores and retrieves information associated with parameter probability density functions (PDFs). PDFs are read from the "tpa.inp" file during run time. The number of distributions includes: CONSTANT, UNIFORM, LOGUNIFORM, NORMAL, LOGNORMAL, BETA, LOGBETA, TRIANGULAR, LOGTRIANGULAR, and EXPONENTIAL. In addition joint PDFs relating two parameters will be supported using a correlation matrix approach. All PDFs will be sampled for each of the realizations required in the simulation.

4.3.3 Invent

INVENT is a utility module that allows centralized computation and storage of radionuclide inventory data. This module is a set of subroutines based on the subroutines described in Lozano et al. (1994). The subroutines provide the inventory (in Ci/MTU) of 43 radionuclides for times to one million years.

4.3.4 Subarea

SUBAREA is a utility module for the storage and retrieval of repository SA information. The database is created once in the READER module and then the information will be available to all other modules. The consequence modules can acquire information about the SA discretization, but not change the information.

4.4 CONSEQUENCE MODULES

The main consequence modules in the TPA code are UZFLOW, NFENV, EBSFAIL, FAULTING, SEISMO, VOLCANO, EBSREL, UZFT, SZFT, ASHPLUME, DCAGW, and DCAGS. These modules will interface with EXEC using a subroutine call statement. The parameters and arrays being passed in the call statement are negotiated between the EXEC and each of the modules. Consistent with the software design principle of procedural abstraction, the EXEC does not need to know how the calculations are performed in the consequence modules. There are at least four ways of doing the calculations: abstraction of results (i.e., table look-up), abstraction of models, incorporation of the main calculational routines from an existing code, or spawning an independent process that executes a stand alone code. Previously, the EXEC explicitly called routines and spawned processes. In the new EXEC, main consequence modules are called directly and the I/O between EXEC and modules need not be changed in the future if the implementation in the module changes.

Because the TPA code simulates the time-dependent response of the proposed repository, it decides the overall time discretization to be used for all consequence modules. The time discretization is intended to synchronize input and output between modules and should not be confused with timesteps used in solving transient problems in various modules. The time discretization can and will often be nonuniform, especially for simulations out to hundreds of thousands of years. An example of a time discretization is {0, 10, 25, 50, 75, 100, 125, 150, 200, 250, 300, ..., 9,500, 10,000} yr. All of the time-dependent inputs and outputs for the consequence modules must be provided at these time steps.

The UZFLOW module will provide estimates of percolating flow rates into the near-field of the proposed repository. Separate flow rates will be estimated for each of the proposed repository SAs. For example, if six SAs are used for the proposed repository, then UZFLOW will be called six times to provide estimates of flow into the near-field. The flow rates are time-dependent and need to be predicted for the times provided by the TPA code. This module may account for long-term trends that affect percolation (e.g., climate changes) or short-term changes (e.g., abnormal wet period). Output data from the UZFLOW module will be input to the NFENV module.

The NFENV module will provide estimates of near-field conditions for each proposed repository SA. The NFENV module should account for the location of the SA (interior or edge regions). The output of the NFENV module is the near-field rock temperature, WP surface temperature, spent fuel temperature, relative humidity at the WP, flow rates into the EBS, and geochemical condition of groundwater flowing into the EBS. All of this output will be time-dependent. These data will be provided to EBSFAIL module.

The EBSFAIL module uses the output of the NFENV module to predict failure of the EBS to contain waste. Failure can be the result of corrosion, as well as other anticipated causes. Examples of anticipated causes of failure include initial defects, thermal-induced stresses in the WP, and anticipated seismic activity. A separate SEISMO module will also evaluate the consequences of seismic activity. At this time, EBSFAIL should account for the numerous small magnitude events while SEISMO should evaluate low-probability, high-consequence events. Possibly, EBSFAIL should evaluate activity with recurrence intervals of up to 500 – 1,000 yr, and SEISMO should evaluate stronger events that have longer recurrence intervals. EBSFAIL and SEISMO analysts need to negotiate this detail. Low-probability, high-consequence disruptive causes of failure are not considered in EBSFAIL. The primary output of EBSFAIL will be a time-dependent fraction of EBS failure to contain the waste. The fraction may start at a nonzero value due to initial defects and may not reach 100 percent within maximum simulation time.

The FAULTING, SEISMO, and VOLCANO modules each predict failure of the EBS due to disruptive processes and events or additional mechanical loads for the WP that accelerate failure of the EBS. These modules are called only for disruptive scenarios and not for the base-case scenario. Each module generates a time-dependent failure curve for WPs in each realization. These failures are combined by the TPA code with the EBSFAIL failures to have an overall percent failure.

The EBSREL module predicts the transient release rate [Ci/yr] of each radionuclide per WP in the SA. The radionuclides are released from the EBS and into the lower unsaturated zone region that extends from below the proposed repository to above the water table.

The UZFT module predicts release [Ci/yr] of each radionuclide from the unsaturated zone into the saturated zone. The module simulates gravity-driven percolating flow and radionuclide transport in

the fractured, stratified hydrogeology. The module accounts for the retardation of nuclides. The sum of the releases from all SAs is then provided to the SZFT module.

The SZFT module predicts the transient groundwater source [Ci/m³] of each radionuclide at the CP which may be a well located 5, 20, 25 or 30 km away from the proposed repository. The module accounts for the retardation of nuclides, plume dilution, and dilution due to the pumping rate at the well.

The ASHPLUME module provides an extra pathway for radionuclides to be transported into the biosphere. This pathway is due to extrusive volcanic events that entrain waste in the magma and spread the waste in the volcanic ash plume. After VOLCANO is called, ASHPLUME will be called to evaluate this pathway for waste.

The DCAGW module simulates the biosphere and computes dose rates [rem/yr] from the groundwater pathway. DCAGW uses the output of SZFT. The DCAGW module is based on the GENII code which has been applied to YM biosphere conditions.

The DCAGS module simulates the biosphere and computes annual doses [rem/yr] from ground surface pathways. DCAGS uses output from ASHPLUME. The DCAGS module is based on the GENII code which has been applied to YM biosphere conditions.

4.5 OUTPUT FROM TOTAL PERFORMANCE ASSESSMENT CODE

The TPA code will generate results that can be used in importance analyses and in assessment of proposed repository compliance with either dose, risk, or release-based standards. The output files generated are listed here:

| | |
|-------------|--|
| sp.dat | = input values sampled for each R from PDFs described in "tpa.inp" |
| mv.dat | = constant (e.g., not time- or nuclide-dependent) module variables for each module for each R/SA that will be used for importance analyses |
| uzflow.dat | = output values for each R/SA from the UZFLOW module |
| nfenv.dat | = output values for each R/SA from the NFENV module |
| ebsfail.dat | = output values for each R/SA from the EBSFAIL module |
| ebsrel.dat | = output values for each R/SA from the EBSREL module |
| uzft.dat | = output values for each R/SA from the UZFT module |
| szft.dat | = output values for each R from the SZFT module |
| dcagw.dat | = output values for each R from the DCAGW module |
| dcags.dat | = output values for each R from the DCAGS module |

77/95

doseavg.dat = annual effective dose equivalent at CP

ccdf.dat = data used to generated complementary cumulative distribution function for Environmental Protection Agency normalized release to accessible environment located at 5 km from the proposed repository over a 10,000 yr time period

The first two lines of any output file will echo the first two lines of the "tpa.inp" file. The third line of any output file will provide the version number of the TPA code being used and the time and date of the run. Data will follow on subsequent lines depending on the specific file.

5 PROGRAMMING LANGUAGES

The TPA code is written in FORTRAN 77 as implemented in the SUN SPARCompiler, Version 2.0. The length of variable names will not be restricted to seven or less characters, but will be as long as needed to readily identify the variable. In addition, some compiler specific calls for date and time will be used. Although not recommended, modules can be written in languages other than FORTRAN 77 or can be standalone computer programs. In these cases, the responsible programmers must provide a FORTRAN 77 interface subprogram consistent with the TPA-module interface described in the previous section.

6 HARDWARE PLATFORMS

The TPA code will be developed for execution on SUN machines using the UNIX operating systems. The code will be designed such that it will also run on other operating systems to the extent practical, such as the CRAY computer.

7 GRAPHICS OUTPUT

No special graphics devices will be supported. Output will be in the form of ASCII files written in a format that can be read by spreadsheet programs, analysis software, and plotting packages.

8 PRE AND POST-PROCESSOR

No pre or post-processor is required or supported by the TPA code. The output files generated by the TPA code will be designed so that they can be read easily by spreadsheet programs, analysis software, and plotting packages.

9 REFERENCES

- Lozano, A.S., H. Karimi, J.P. Cornelius, R.D. Manetufel, and R.W. Janetzke. 1994. *INVENT: A module for Calculation of Radionuclide Inventories, Software description, and User Guide*, CNWRA 94-016. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.
- Sagar, B., and R.W. Janetzke. 1993. *Total-System Performance Assessment (TPA) Computer Code: Description of Executive Module, Version 2.0*. CNWRA 93-017. San Antonio, TX: Center For Nuclear Waste Regulatory Analyses.

78/95

Rationale for Changes from TPA Version 2 to TPA Version 3 Computer Code

Background:

The NRC TPA computer code (Sagar and Janetzke, 1993), designated version 2, was developed in 1993 for the NRC IPA Phase 2 Assessment (Wescott et al., 1994) of the proposed YM repository site. The TPA version 2 code was designed to calculate the performance measures specified in NRC 10 CFR 60 (i.e., container lifetime, release rate, groundwater travel time) and the EPA standard (i.e., cumulative release for 10,000 yrs). Additionally, the code calculates integrated population dose over 10,000 yrs. Since the initial development, there have been no major upgrades to the TPA version 2 code (although attempts were made to improve the code computational efficiency). NUREG-1464 (Wescott et al., 1994) documents recommendations for improvement of the TPA version 2 code including: (i) software QA requirements need more prominence in module development, (ii) future IPA developments will require more model abstractions and efficient computing techniques, and (iii) the TPA code must be easily upgraded. With regards to the latter recommendation, NUREG-1464 states that the TPA system code "be considered a dynamic entity, to be upgraded in future IPA iterations. Possible upgrades include: addition of new modules, changed scope of current modules, centralized use of databases, uniform interfaces between modules, and uniform coding practices among modules." Most of these recommendations have been followed in developing TPA version 3, particularly those related to software QA.

Major revisions of the TPA code were required to address: (i) recommendations of the National Academy of Sciences (which will require the EPA to establish new performance criteria), (ii) changes in the DOE repository design, (iii) expanded knowledge base, improved models, and additional data compiled by NRC/CNWRA, and (iv) specific needs for the KTI sensitivity analyses. In addition, recent applications of the TPA version 2 identified certain software QA vulnerabilities and some problems with regard to the lack of flexibility of the code to accommodate new requirements and to allow simple modifications.

The recommendations of the NAS have required the following changes in Total-System Performance Assessment (TSPA) methodology for the proposed repository: (i) addition of a new performance measure (individual dose versus cumulative release), (ii) addition of variable compliance period (time to peak dose, not necessarily 10,000 yrs), (iii) elimination of human intrusion by drilling (not considered part of the main compliance determination), and (iv) incorporation of a more appropriate representation of environmental pathways and dose.

DOE repository design changes that necessitated TPA code modifications include waste package design (e.g., in-drift emplacement of large packages versus vertical emplacement of small packages) and areal heat loading (which determines the area requirements and layout of the repository blocks). The NRC technical assistance studies and former research projects have produced technical bases for improving various modules of the TPA code. Some of the major technical contributions are:

- The former Container Lifetime and Source Term (CLST) KTI produced an improved source term module.
- Igneous Activity (IA) KTI has produced improved technical basis for the probability model and scenario characteristics for simulating volcanic eruptions.

- Structural Deformation and Seismicity (SDS) KTI has produced an improved hydrostratigraphic model and technical specifications for a new consequence module to simulate a faulting scenario.
- Thermal Effects on Flow (TEF) KTI has provided improved predictions of temperature and relative humidity (for use in the source term module).
- Unsaturated and Saturated Flow under Isothermal Conditions (USFIC) KTI has produced improved information on climate and the distribution of infiltration over the repository area.

In the NRC/CNWRA audit review (Baca and Brient, 1996) of the DOE TSPA-95 report, the TPA Version 2 code was run to independently check selected DOE CCDFs for cumulative release. In this application, an attempt was made to modify the TPA Version 2 code to accommodate compliance periods longer than 10,000 yrs. It was found that this simple extension of time could not be accommodated without significantly modifying the source term module as well as other modules. Additional code limitations and problems were also identified. For example, it was found that certain input data to one module was not used because of undocumented "hard coding" in the module, the LHS module could not accommodate the beta (or log-beta) distribution used by DOE, and an error in units conversion was found in another module. Other limitations of the TPA version 2 code were associated with duplication of data files (creating potential consistency problems), numerous large data files created by the code (requiring computers with large disk space), and very long run times for individual scenarios.

Transition from TPA Version 2 to TPA Version 3:

The TPA Version 2 executive and modules, which were developed in 1993, are documented in Sagar and Jantezke (1993), Wescott et al. (1994) and in various consequence module users guides. The TPA executive is structured to control the sequence of execution of individual modules. Those modules consist of LHS (random parameter sampler), CANT2 (waste package temperature simulator), FLOWMOD (unsaturated zone flow), C14 (carbon-14 gas phase transport), SOTEC (source term code), NEFTRAN (radionuclide transport), CLIMATO (place holder for climate dependent infiltration rate), SEISMO (seismic impact on waste packages), DRILLO1 and DRILLO2 (human intrusion consequences), VOLCANO (volcanism consequences), AIRCOM (interface for airborne releases), and DITTY (population dose module).

The TPA version 3 Software Requirements Description (SRD) (Manteufel, 1997) provides detailed descriptions of the major code changes necessary to address the new performance criteria, design characteristics, and incorporate improved models. To ensure that the planned KTI sensitivity studies could be completed in a timely manner, it was considered vital that the new version of the TPA code be very transparent (i.e., easy to understand and follow the coding logic), have relatively short run times, and permit easy to access intermediate and final calculations. The following descriptions summarize the major revisions to the individual TPA modules and the technical reasons for the changes.

TPA Executive:

The TPA Version 2 executive or driver module was rewritten to: (i) provide greater transparency of the flow of data and calculational results between modules, and (ii) permit the user to have direct access to inputs and outputs for the conduct of sensitivity analyses. The TPA version 3 executive is about

500 lines of Fortran code versus the 20,000 lines of code in TPA version 2. The new version retains the TPA Version 2 algorithm for computing the CCDF, but uses a different approach for invoking the sequence of consequence module runs (i.e., subroutine calls rather than spawning processes).

LHS Sampler:

The new Sampler module retains the same sampling approach of the original LHS module but with the added capability for additional distributions (beta and log-beta) which are commonly used by DOE. In addition, the new module removes the ordering restriction on input parameters and enhances the transparency of parameter transfers from the sampler to the consequence modules.

CANT2 Module:

The CANT2 module will be upgraded by incorporating a 3D conduction only model. This upgraded version is being developed under the TEF KTI. In addition, time histories of thermal-chemical processes will be generated with MULTIFLO and added at a later time.

FLOWMOD Module:

Functionality of the FLOWMOD module will be retained in the new module called UZFLOW, and modified to reflect more recent information regarding infiltration distributions provided by USFIC KTI.

C14 Module:

This module has been eliminated because gas phase carbon-14 is not significant from an individual dose standpoint. TSPAs conducted by NRC and DOE have shown that the numerical limit for C-14 cumulative release (in the EPA 1985 standard) is overly stringent and inappropriately constrain the performance of the repository.

SOTEC Module:

This module has been replaced because of the availability of the new EBSPAC modules (EBSFAIL and EBSREL) which contains the most recent waste package design and improved models for container life and release calculations. The NRC staff have assumed responsibility for further development of the EBSREL module.

NEFTRAN Module:

This module has been retained with minor modification and is utilized in the new UZFT and SZFT modules. In consultation with the NRC staff, the module input/output and hydrostratigraphic data are currently being modified for incorporation in the new version of TPA.

CLIMATO Module:

This place holder module (renamed CLIMATE) will be used to incorporate a description of long-term changes in infiltration rates. The data and calculational approach will be provided by the USFIC KTI.

SEISMO Module:

This module will be superseded by an upgraded module currently being developed under the TSPA KTI (transferred from the former RDTME KTI). The development of the new module is required to accommodate the new waste package design and incorporate new models of seismicity and drift stability.

DRILLO1 and DRILLO2 Modules:

These two modules were eliminated because human intrusion by drilling will not be considered in compliance determinations, as recommended by the NAS. In addition, previous consequence analyses (see Annual Report) have indicated that this scenario has little impact on dose and cumulative release.

VOLCANO Module:

This module has been modified to incorporate the probability map and scenario characteristics provided by the IA KTI. In addition, the module was linked to a new ASHPLUME module that calculates ash/radionuclide dispersal from the simulated volcanic eruption.

AIRCOM Modules:

This module, which interfaces with DITTY, will be eliminated because it no longer required.

DITTY Modules:

This module will be eliminated because the population dose is no longer a relevant performance indicator (per the NAS recommendations for individual dose). DITTY will be replaced by look-up table modules DCAGW (dose factors for groundwater pathway) and DCAGS (dose conversion factors for ground surface pathways). Dose factors for DCAGW and DCAGS will be computed using the GENII dose code.

New TPA Modules and Utilities:

Three new modules have been added to permit consideration of relevant features, events, and processes. These new modules include: (i) FAULTING (calculates the impact of fault displacements on waste packages), (ii) ASHPLUME (calculates the ash and radionuclide dispersal resulting from a volcanic eruption through the repository), and (iii) NFENV (provides parameter specifications for the near-field environment). Various utility routines were developed to centralize data sets and calculational algorithms. For example, the INVENT utility module was developed to provide a centralized data set for the radionuclide inventory in the waste package as a function of time and spent fuel characteristics.

Summary:

The basic rationale for revising the TPA code was: (i) the need for a code suitable for use by a wide variety of users, (ii) the requirement to accommodate expected changes in the regulations (e.g., performance measures, compliance time period, etc), (iii) repository design changes (waste package design and emplacement geometry), and (iv) the need to incorporate the new and improved modules. Because the TPA code will now be used by many users, it was deemed vital that new version of the code

be highly transparent (with regards to its functionality), significantly easier to modify and use, and computationally more efficient. In developing the TPA version 3, the existing consequence modules that were relevant to the new requirements were placed under a new and more compact executive program. The consequence and utility modules that have been developed/modified to-date have been extensively tested for software QA purposes. Many of the improvements to the TPA code are consistent with the recommendations outlined in NRC IPA Phase 2 report (Wescott, et al., 1994). The design of the new TPA version 3 code is documented in a series of draft SRDs (Manteufel, 1997) which were provided to and discussed with the NRC staff, in meetings such as the one on November 11, 1996 (prior to the Annual Center Review meeting).

References:

- Baca, R.G., and R.D. Brient (Editors). 1996. *Total System Performance Assessment 1995 Audit Review*. Letter report submitted to NRC. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.
- Manteufel, R.D. 1997. *Total Performance Assessment (TPA) Code Version 3.0 Software Requirements Description*. Letter report submitted to NRC. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.
- Sagar, B., and R. Janetzke. 1993. *Total-System Performance Assessment (TPA) Computer Code: Description of Executive Module, Version 2.0, Iterative Performance Assessment, Phase 2*. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.
- Wescott, R.G., M.P. Lee, N.A. Eisenberg, T.J. McCartin, and R.G. Baca (Editors). 1994. *NRC Iterative Performance Assessment Phase 2*. NUREG-1464. Washington, DC: U.S. Nuclear Regulatory Commission.

[63] From: Randall Manteufel at CNWRA 12/6/96 4:16PM (1096 bytes: 1 ln)
To: Bruce Mabrito at CNWRA-OS2
cc: Mark Jarzemba, Amitava Ghosh at CNWRA-SUN, Robert Baca at CNWRA-OS2
Subject: SRD's

----- Message Contents -----

Bruce,

We have decided to develop stand-alone SRDs for the ASHPLUME & FAULTING computer programs given their
1) extensive use at the Center, 2) recognition by the NRC staff, and 3) anticipated future use in Center deliverables.

Earlier, we had planned to include descriptions of these 2 codes in the TPA SRD. But now we will issue individual SRDs.

You will receive 3 SRDs for the following codes:

- 1) FAULTING (lead: Amit Ghosh) by 12/15/96
- 2) ASHPLUME (lead: Mark Jarzemba) by 12/15/96
- 3) TPA 3.0 (lead: Randy Manteufel) by 1/15/97

Individual User Guides will also be completed for each of the 3 codes. Anticipated dates are:

- 1) FAULTING (lead: Amit Ghosh) by 1/20/97
- 2) ASHPLUME (lead: Mark Jarzemba) by 2/97 (?)
- 3) TPA 3.0 (lead: Randy Manteufel) by 3/17/97

Thanks.

-Randy

Total-System Performance Assessment (TPA) Code Software Requirements Description

Introduction

This Software Requirements Description (SRD) document is the first step in updating the Total-system Performance Assessment (TPA) code from version 2.0 to 3.0. The TPA code version 2.0 was used in the NRC/CNWRA Iterative Performance Assessment (IPA) phase 2 exercise. The TPA code is a probabilistic driver for a set of consequence modules that simulate the performance of a geologic repository of nuclear High-Level Waste (HLW) at Yucca Mountain (YM), NV. The TPA code consists of the executive module that controls the consequence modules, a set of utility modules, and a set of discipline-specific consequence modules that simulate major safety components of the repository system. The TPA code integrated geologic site characterization data, repository and waste package engineered designs, and biosphere data. The consequence modules are designed with input from experts such as materials engineers, hydrogeologists, seismologists, volcanologists, and health physicists. Recent developments in the proposed YM repository necessitate a new version of TPA. These developments include (i) new repository, waste package and emplacement designs, (ii) changing regulatory standards (from release-based to dose-based), and (iii) potentially longer time periods of concern (potentially up to hundreds of thousands of years). Other numerous improvements will be incorporated that reflect knowledge and data gained in recent years of site characterization and laboratory studies as well as other total system performance assessments.

1. Software Function

The TPA code is a probabilistic driver that integrates a set of deterministic consequence modules to perform an overall assessment of the performance of the proposed YM HLW repository. The TPA code controls the probabilistic sampling of input parameters, the calculational flow process between modules, and the generation of output files. Output files can be used for parameter importance analyses, generation of time-dependent risk curves, and generation of complementary cumulative distribution functions for cumulative release of radionuclides. The TPA code has overall responsibility for the description of the repository system and controlling the flow of execution and data exchange between consequence modules. For example the TPA code will control the spatial and temporal discretizations (i.e., sub-area discretization of the repository in and time stepping scheme). In the NRC/CNWRA Iterative Performance Assessment (IPA) Phase 2 exercise, there were 7 sub-areas and 50 time steps of 200 yr. The number of sub-areas (SA) in TPA can be based on the latest repository design and reflect near-field thermal-hydrologic-mechanical-chemical (THMC) environments in the repository as well as hydrostratigraphy. The time stepping will be variable over the simulation as well as the total time period of interest.

2. Technical and Computational Approach

The overall conceptual approach used in a total system performance assessment (TSPA) as implemented in the TPA code, is outlined in Figure 1. Data flows from the system characterization to final regulatory compliance determination (shown as either a cumulative release or dose based standard). The bulk of the modeling effort is in the consequence modules which includes both anticipated processes (also called base-case processes) and disruptive processes. The base-case system has 7 major subsystem models:

- ground water flow from the ground surface to the repository,
- near-field thermal-hydrologic-mechanical-chemical environment of the engineered barrier system (EBS),
- corrosion and other anticipated failure mechanisms of the EBS containment,
- release of radionuclides from the EBS and into geologic setting (GS),
- ground water flow and radionuclide transport in the unsaturated zone below the repository and into the saturated zone,
- ground water flow and radionuclide transport in the saturated zone from the repository to a compliance point (CP) or boundary, and the
- transport of radionuclides in the biosphere through the ground water pathway which leads to dose to humans.

The disruptive system has faulting, seismicity, and volcanism which cause earlier failures of EBS containment. In the case of volcanism, radionuclides may be released directly into the accessible environment (AE) and at the CP through the ground surface pathway.

All of the consequence modules will be deterministic. A number of utility modules will be used to provide general data and generic capabilities that more than one module may need. For example, the initial radionuclide inventory will be calculated in a module so this information can be provided to other modules.

The TPA code will control the spatial discretization of the repository (i.e., number of sub-areas), the distance from the repository to the CP (e.g, 5, 20, 25, or 30 km), the temporal discretization scheme (e.g., output every 200 yr), and the maximum time of interest (e.g., 100,000 yr).

Major analysis improvements in TPA include:

- ability to increase or decrease the time period of interest (TPI)
- ability to evaluate finer repository discretizations
- ability to evaluate different areal mass loading (AML)
- ability to calculate time-dependent dose rate at a compliance point (CP)
- ability to calculate peak dose rate at a CP in the TPI
- ability to evaluate dilution in saturated zone (SZ)
- ability to evaluate alternative EBS designs

- improved ability for importance analysis
- updated consequence models
- streamlined scope for consequence modules
- streamlined system for data transfer between consequence modules
- more flexible design to accommodate changes in consequence modules

3. User Interface and Data Flow

The TPA code will be executed in file batch mode using one main input file call tpa.inp. The TPA code reads data from only this one input file. The TPA code writes output data into a set of files for plotting or importance analyses. The specific file interfaces are described here.

3.1 Input to TPA Code

All of the input data that dictate the type of analysis is contained in the "tpa.inp" file. No other files/input will have an effect on the results of the calculation. The tpa.inp file contains all of the parameters necessary to describe the scenario and the number of realizations being requested. The tpa.inp file starts with 2 comment lines that the analysts should use to describe the type of run being performed. These lines will be read as Character*80 and echoed at the tops of all output files.

3.2 Flow of Data between Modules

The TPA code controls data flow by passing data in the subroutine call statement to each module. The TPA code does not use common blocks or disk files for data transfer between itself and consequence modules. The TPA code also does not permit that data be passed directly between consequence modules. Each module is to be called only by the TPA code, and not by other modules. For efficiency and implementation purposes, the modules can consist of more than one subroutine, may call TPA utility subroutines (e.g., INVENT), or may call stand alone programs (e.g. NEFTRAN). But modules are to pass information only to the TPA code in order to control the simulation.

The overall sequence of execution and flow of data is shown in Figure 2. Here, the TPA code starts the simulation by reading the tpa.inp file through the READER routine. The READER module need only be called once during a run. Having determined the parameters that describe the system, the TPA continues by calling component specific consequence modules. Some modules will only be called once during one realization, while others will be called many times. Modules being called once include SAMPLER, SZFT, DCAGW and DCAGS. The modules UZFLOW, NFENV, EBSFAIL, EBSREL, and UZFT will be called once for each SA for each R. If disruptive scenarios are being analyzed, the FAULTING, SEISMO, VOLCANO, ASHPLUME and DCAGS modules will be called directly by the TPA code once during a realization. These disruptive modules will be used to either cause earlier failure of the EBS, or provide a more direct pathway for radionuclides into the biosphere (e.g., VOLCANO through

ASHPLUME). If desired in the future, disruptive scenarios can be designed or modified to affect ground water flow and the transport of radionuclides.

The computational scheme for TPA is shown in Figure 3. Each module is called by TPA with some inputs and returns some outputs to TPA through the call statement. Consequence modules do not call each other directly. There is a clear expectation of inputs and outputs between TPA and each module. In most cases, the output of one module is the input to the next module. TPA has two main loops for the number of realizations and the number of sub-areas.

The utility modules and consequence modules are discussed next.

3.3 Utility Modules

3.3.1 SAMPLER

SAMPLER is a utility module that pre-processes the data from the tpa.inp file. This is the only subroutine that reads the tpa.inp file. This module is similar to that already existing in the TPA version 2.0 code. This module can interface with the existing LHS program to implement the Latin Hypercube Sampling (LHS) method or Monte Carlo random sampling. All of the input parameters will be read from the tpa.inp file and processed once for all of the realizations required in the simulation.

3.3.2 INVENT

INVENT is a utility module that allows the centralized computation and storage of radionuclide inventory data. This module consists of a set of subroutines that are based on the subroutines described in CNWRA 94-016. The subroutines provide the inventory (in Ci/MTU) of 43 radionuclides for times out to 1 million years. Each subroutine in this module are described in Appendix.

3.3.3 ARRAY

ARRAY is a utility module that groups processes on arrays of data (also called vectors of data). The module contains sorting, calculation of mean and variance, and sampling at quantile level capabilities. Each subroutine is described in the Appendix

3.3.4 SUBAREA

SUBAREA is a utility module for the storage and retrieval of repository sub-area information. The database is created once and then the information can be acquired by a number of different calls.

3.4 Consequence Modules

The main consequence modules in the TPA code are UZFLOW, NFENV, EBSFAIL, FAULTING, SEISMO, VOLCANO, EBSREL, UZFT, SZFT, ASHPLUME, DCAGW, and DCAGS. These modules will interface with the TPA code.

Because the TPA code simulates the time-dependent response of the repository, it decides the overall time discretization that will be used for all consequence modules. The time discretization can and will often be non-uniform, especially for simulations out to hundreds of thousands of years. An example of a time discretization is: {0, 10, 25, 50, 75, 100, 125, 150, 200, 250, 300, ..., 9500, 10000} yr. All of the time-dependent inputs and outputs for the consequence modules must be provided at these time steps.

The UZFLOW module will provide estimates of percolating flow rates into the near-field (NF) of the repository. Separate flow rates will be estimated for each of the repository sub-areas. For example if 6 sub-areas are used for the repository, then UZFLOW will be called 6 times to provide estimates of flow into the NF. The flow rates are time-dependent and need to be predicted for the times provided by the TPA code. Output data from the UZFLOW module will be input to the NFENV module.

The NFENV module will provide estimates of near-field conditions for each repository SA. The NFENV module should account for the location of the SA (interior or edge regions). The output of the NFENV module is the near-field rock temperature, waste package (WP) surface temperature, spent fuel temperature, relative humidity (RH) at the WP, flow rates into the EBS, geochemical condition of ground water flowing into the EBS. All of this output will be time-dependent. This data will be provided to EBSFAIL module.

The EBSFAIL module uses the output of the NFENV module to predict the failure of the EBS to contain the waste. Failure can be the result of corrosion as well as other anticipated causes of failure. Examples of anticipated causes of failure include initial defects, thermal-induced stresses in the WP, and anticipated seismic activity. A separate SEISMO module will also evaluate the consequences of seismic activity. At this time, EBSFAIL should account for the numerous small magnitude events while SEISMO should evaluate low-probability, high consequence events. Possibly, EBSFAIL should evaluate activity with recurrence intervals of up to 500 to 1000 years, and SEISMO should evaluate stronger events that have longer recurrence intervals. This detail will need to be negotiated by the EBSFAIL and SEISMO analysts. Low-probability, high-consequence disruptive causes of failure are not considered in EBSFAIL. The primary output of EBSFAIL will be a time-dependent fraction of EBS failure to contain the waste. The fraction may start at a non-zero value due to initial defects, and may not reach 100% within the maximum simulation time.

The FAULTING, SEISMO, and VOLCANO modules each predict failure of the EBS due to

disruptive processes and events. These modules are called only for disruptive scenarios and not for the base-case scenario. Each module generates a distribution of percent WP failures at each time in the simulation. These failures are combine by the TPA code with the EBSFAIL failures to have an overall percent failure.

The EBSREL module predicts the transient release rate [Ci/yr] of each radionuclide per WP from the EBS. The radionuclides are released from the EBS and into the lower UZ region that extends from below the repository to above the water table.

The UZFT module predicts the transient release rate [Ci/yr] of each radionuclide into the saturated zone. The module simulates gravity-driven percolating flow and radionuclide transport in the fractured, stratified hydrogeology. The module accounts for the retardation of nuclides. The sum of the releases from all columns is then provided to the SZFT module.

The SZFT module predicts the transient ground water source [Ci/m³] of each radionuclide at the CP which may be a well located 5, 20, 25 or 30 km away from the repository. The module accounts for the retardation of nuclides, plume dilution, and dilution due to the pumping rate at the well.

The ASHPLUME module provides an extra pathway for radionuclide to be transported into the biosphere. This pathway is due to extrusive volcanic events which entrain waste in the magma and spread the waste in the volcanic ash plume. After VOLCANO is called, ASHPLUME will be called to evaluate this pathway for waste.

The DCAGW module simulates the biosphere and computes dose rates [rem/yr] from the ground water pathway. DCAGW uses the output of SZFT. The DCAGW module is based on the GENII code which has been applied to YM biosphere conditions.

The DCAGS module simulates the biosphere and computes dose rates [rem/yr] from ground surface pathways. DCAGS uses output from ASHPLUME. The DCAGS module is based on the GENII code which has been applied to YM biosphere conditions.

3.4 Output from TPA Code

The TPA code will generate results that can be used in importance analyses, and assessment of repository compliance with either dose, risk, or release based standards. The output files generates are listed here:

xin.tpa.dat = input values sampled for each R from probability distribution functions (PDFs) described in tpa.inp

xmod.tpa.dat = output values stored for each module for each R/SA that will be used for importance analyses.

uzflow.tpa.dat = output values for each R/SA from the UZFLOW module
 nfenv.tpa.dat = output values for each R/SA from the NFENV module
 ebsfail.tpa.dat = output values for each R/SA from the EBSFAIL module
 ebsrel.tpa.dat = output values for each R/SA from the EBSREL module
 uzft.tpa.dat = output values for each R/SA from the UZFT module
 szft.tpa.dat = output values for each R from the SZFT module
 dcagw.tpa.dat = output values for each R from the DCAGW module
 dcags.tpa.dat = output values for each R from the DCAGS module
 doseavg.tpa.dat = average time-dependent dose rate at CP
 ccdf.tpa.dat = data used to generate Complementary Cumulative Distribution Function (CCDF) for EPA normalized release to accessible environment located at 5 km from repository over 10,000 yr time period

The first 2 lines of any output file will echo the first 2 lines of the tpa.inp file. The third line of any output file will provide the version number of the TPA code being used, and the time and date of the run. Data will follow on subsequent lines depending on the specific file.

4. Programming Languages

The TPA code is written in FORTRAN 77 as implemented in the SUN SPARCompiler, Version 2.0. The length of variable names will not be restricted to seven or less characters, but will be as long as needed to readily identify the variable. In addition, the executive module will make some compiler specific calls for date and time. Although not recommended, modules can be written in languages other than FORTRAN 77 and/or can be standalone computer programs. In these cases, the responsible coders must provide a FORTRAN 77 interface subprogram that is consistent with the TPA-module interface described in the previous section.

5. Hardware Platforms

The TPA code will be developed for execution on SUN machines using the UNIX operating systems.

9/1/95

6. Graphics Output

No special graphics devices will be supported. Output will be in the form of ascii files written in a format that can be read by spreadsheet programs, analysis software, and plotting packages.

7. Pre- and Post-Processor

No pre or pos-processor is required or supported by the TPA code.

92/95

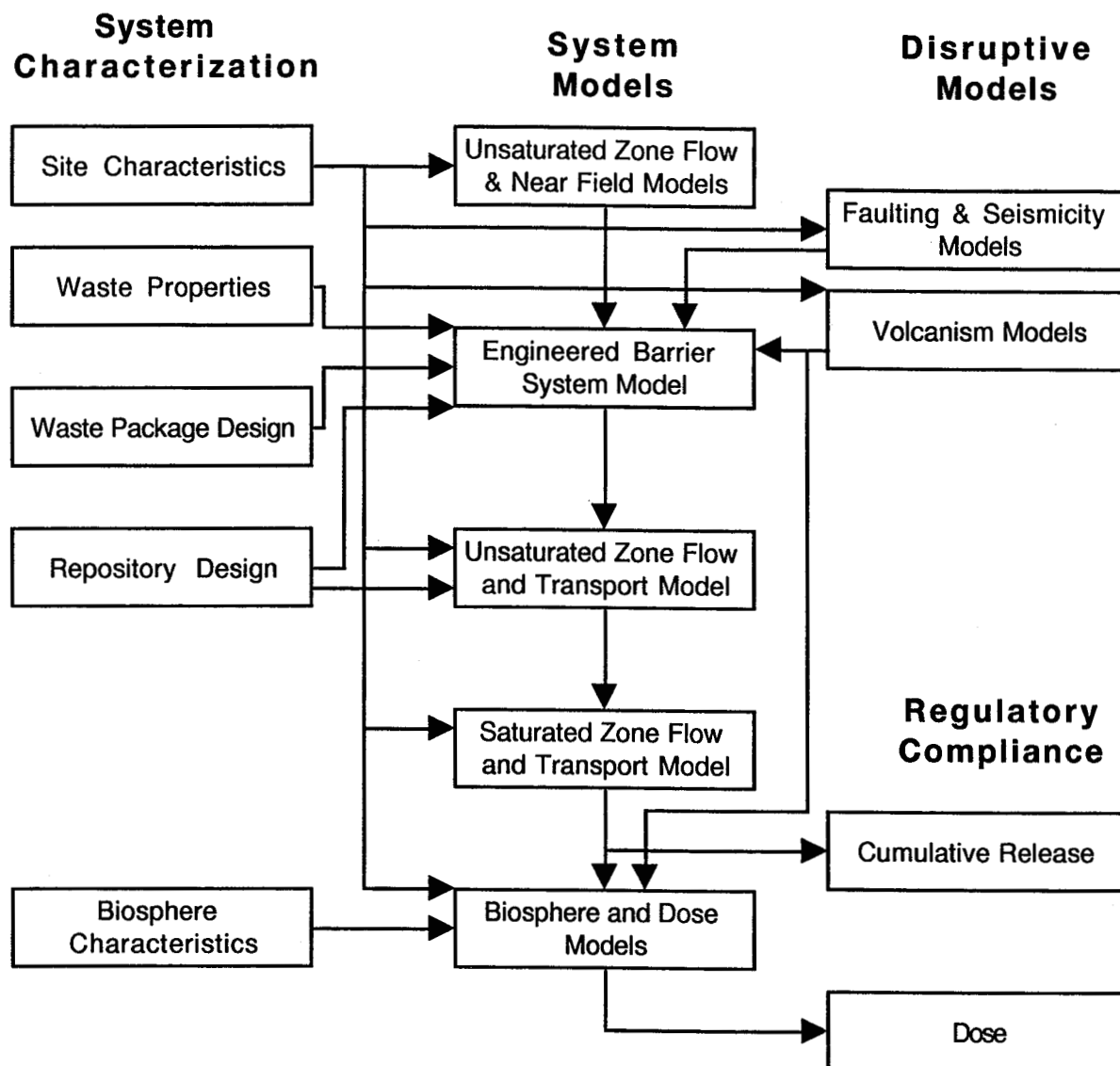


Figure 1 Overall TSPA Flow Diagram

93/95

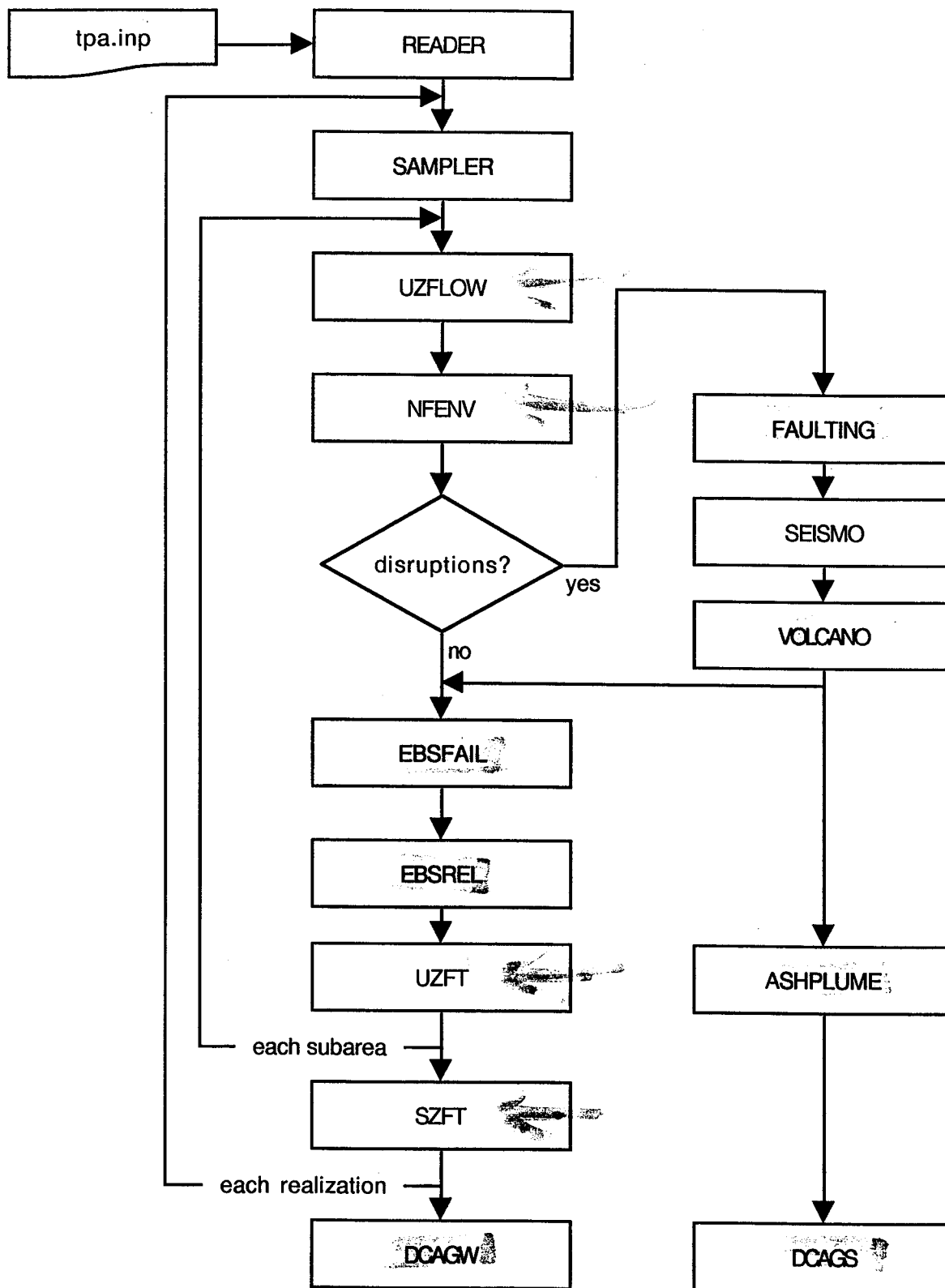


Figure 2. TPA Flow Diagram

94/95

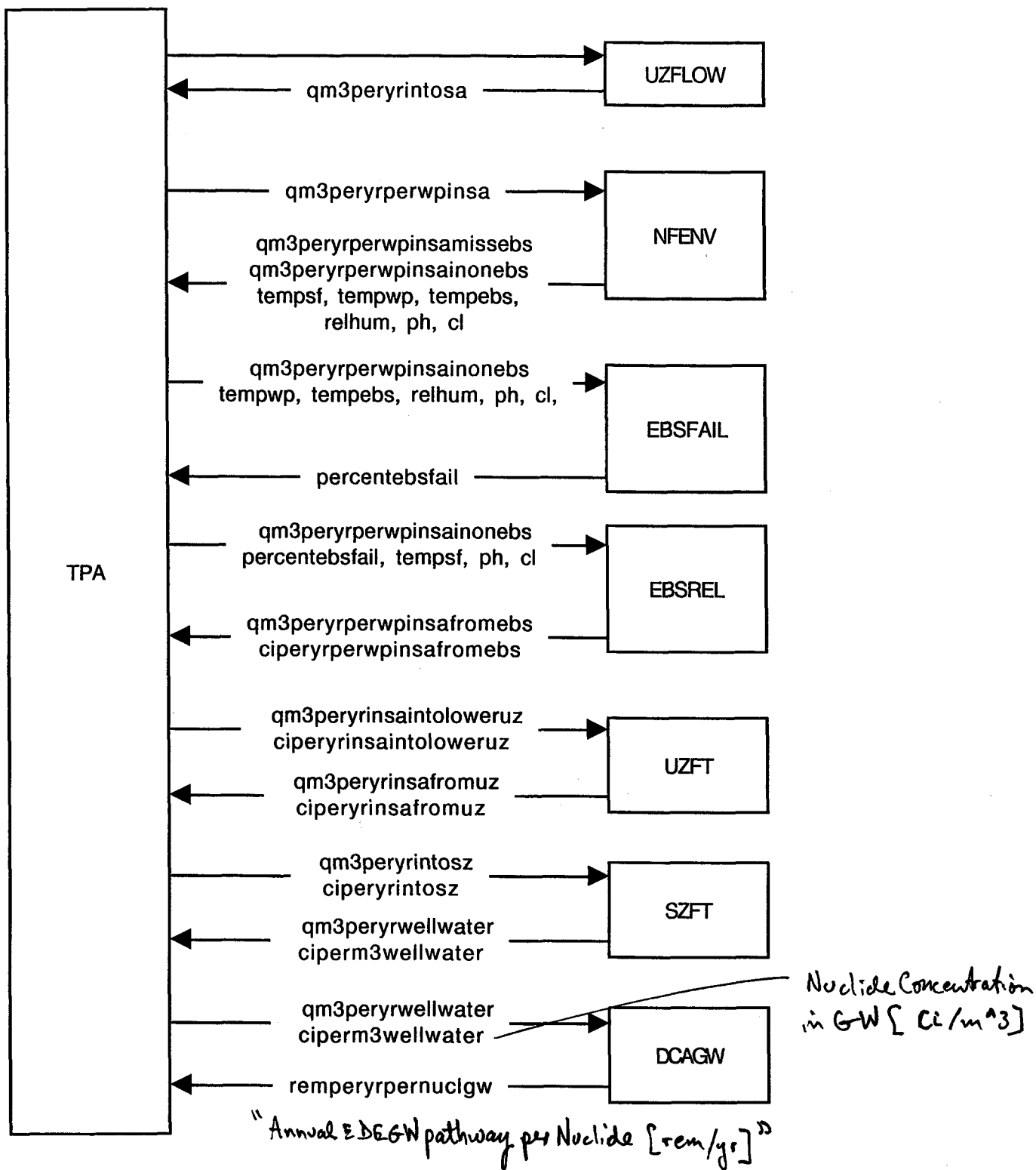


Figure 3. Main Input/Output Associated with Base-Case Flow & Transport

TPA V 3.1

1/100

SOFTWARE RELEASE NOTICE

| | | |
|--|----------------------|--------------------------|
| 1. SRN Number: PA-SRN-205 | | |
| 2. Project Title: TSPA & Technical Integration Code | | Project No. 20-01402-762 |
| 3. SRN Title: TPA Version 3.1 <i>10-4-99</i> | | |
| 4. Originator/Requestor: Ron Janetzke | | Date: Oct. 1, 1999 |
| 5. Summary of Actions <div style="margin-left: 20px;"> <input type="checkbox"/> Release of new software <input type="checkbox"/> Release of modified software: <div style="margin-left: 20px;"> <input type="checkbox"/> Enhancements made <input type="checkbox"/> Corrections made </div> <input type="checkbox"/> Change of access software <input checked="" type="checkbox"/> Software Retirement </div> | | |
| 6. Persons Authorized Access | | |
| Name | Read Only/Read-Write | Addition/Change/Delete |
| Ron Janetzke | RW | |
| Sitakanta Mohanty | RW | |
| Tim McCartin (NRC) | RW | |
| M. Rose Byrne (NRC) | RW | |
| 7. Element Manager Approval: <i>Gordon Wiltmeyer</i> | | Date: <i>10/4/99</i> |
| 8. Remarks: Version superceded. | | |

3/1/97

SOFTWARE SUMMARY FORM

| | | | |
|--|---|---|--|
| 01. Summary Date: 9/5/97 | 02. Summary prepared by (Name and phone) B. Mabrito (210) 522-5149 | 03. Summary Action: New | |
| 04. Software Date: 9/5/97 | 05. Short Title: TPA Version 3.1 | | |
| 06. Software Title: TPA - Total-System Performance Assessment Computer Code, Version 3.1 | | 07. Internal Software ID: None | |
| 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 type="checkbox"/> Scientific/Engineering <input type="checkbox"/> Auxiliary Analyses <input checked="" type="checkbox"/> Total System PA <input type="checkbox"/> Subsystem PA <input type="checkbox"/> Other b. Specific: | |
| 11. Submitting Organization and Address: CNWRA/SwRI 6220 Culebra Road San Antonio, TX 78238 | | 12. Technical Contact(s) and Phone: Sitakanta Mohanty (210) 522-5185 | |
| 13. Narrative: The TPA Code consists of the following modules: UZFLOW, NFENV, EBSREL, UZFT, SZFT, DCAGW, FAULTO, SEISMO, VOLCANO, ASHPLUMO, ASHRMVO, LHS, EXEC. | | | |
| 14. Computer Platform SUN Workstation | 15. Computer Operating System: UNIX | 16. Programming Language(s): FORTRAN | 17. Number of Source Program Statements: approx. 50,000 lines |
| 18. Computer Memory Requirements: 115 MB | 19. Tape Drives: None | 20. Disk/Drum Units: N/A | 21. Graphics: Unknown |
| 22. Other Operational Requirements None | | | |
| 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 | |
| Software Custodian: <i>B. Mabrito</i> Date: 9/5/97 | | | |

4/80

SOFTWARE RELEASE NOTICE

| | | |
|---|------------|----------------------------|
| 01. SRN Number: PA-SRN-151 | | |
| 02. Project Title: TSPA & Technical Integration Code | | Project No. 20-5708-762 |
| 03. SRN Title: TPA Version 3.1 | | |
| 04. Originator/Requestor: Bruce Mabrito | | Date: 9/5/97 |
| 05. Summary of Actions <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Release of new software <input type="checkbox"/> Release of modified software: <ul style="list-style-type: none"> <input type="checkbox"/> Enhancements made <input type="checkbox"/> Corrections made <input type="checkbox"/> Change of access software <input type="checkbox"/> Software Retirement | | |
| 06. Persons Authorized Access | | |
| Name | RO/RW | A/C/D |
| Sitakanta Mohanty | READ/WRITE | ADDED |
| Ron Janetzke | READ/WRITE | ADDED |
| Tim McCartin | READ/WRITE | ADDED |
| M. Rose Byrne | READ/WRITE | ADDED |
| 07. Element Manager Approval: <i>26 Baea</i> | | Date: <i>9/5/97</i> |
| 08. Remarks: TPA Version 3.1 software was sent to NRC Headquarters on 9/5/97. | | |

9/100

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE: TPA Version 3.1

September 5, 1997

Total Performance Assessment (Scientific and Engineering Software) Version 3.1

1. Scientific Notebook Documentation Development: CNWRA Scientific Notebooks numbered: 170 - S. Mohanty and R. Janetzke (EBSFAIL, EBSREL, UZFT, and SZFT); 164 - M. Jarzempa (ASHPLUMO); 202 - S. Mohanty (EXEC, SEISMO, DCAGS, DCAGW, VOLCANO, NFENV, and UTILITY modules); 167 - A. Ghosh (FAULTO); 196 - S. Hsiung (SEISMO); 163 - S. Stothoff (UZFLOW and CLIMATO); 115 - C. Connor and B. Hill (VOLCANO); 190 - R. Janetzke (TPA Version 2.0 and Audit 96); 223 - J. Bogan (General Code Testing). These scientific notebooks document the individual software development, along with the 100-plus Software Problem/Change Reports (SPCRs) that were generated by both the CNWRA and NRC staff working on TPA Version 3.1. Scientific notebooks have been reviewed earlier in the year by the cognizant EMs and were also verified from the QA Scientific Notebook checkout log. TPA Version 3.1 software development was documented in these scientific notebooks.

2. Programming Language: ANSI Standard FORTRAN 77 confirmed by the Software Custodian. See attached hard copies of TPA Version 3.1 pages for examples.

3. Internal Documentation: On 9/4-5/97, B. Mabrito reviewed portions of the TPA Version 3.1 scientific and engineering software on the "Bigbend" SUN Ultra 2 work station. Comments in the subroutine of SEISMO.f were observed by the software custodian. The TPA Version 3.1 software is located on the "Bigbend" and "Scratchy1" servers. The TPA code contained comments and a hard copy page showing representative comments is attached to this Design Verification Report. Additionally, comment segments were checked in the TPA Version 3.1 FAILT.f stand alone and UZFT.f and PEAKFINDER.f subroutines and were found to be clear and numerous. The developer stated that approximately 30-40% of the code is composed of comment lines. The internal documentation comments in TPA Version 3.1 meet the requirements of TOP-018, Section 5.4.4.

4. Software Labels and Data

a. Header Data and Format: TPA Version 3.1 header data and the format were compared against TOP-018 Section 5.4.6 and found acceptable (see several of the attached hard copy pages). See attached sheets.

b. NRC Data: TPA Version 3.1 NRC data and the format were compared against TOP-018, Section 5.4.6, third bullet and found to be acceptable. The disclaimer statements have been added where they were missing in TPA Version 3.0.

c. Source Code Header: TPA Version 3.1 header data was compared to TOP-018 Section 5.4.6, fourth bullet, and found to meet the requirements and are acceptable.

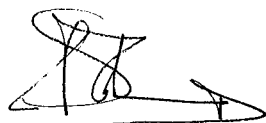
5. Unique Run Identification: At the top of each output file a unique identifier was created on the print out. A page of files was printed and attached to this report to show the following statement: "TPA 3.1beta5, Job started: Thu Sep 4 20:41:58 1997" which fully meets the unique run identification requirements of TOP-018, Section 5.4.5. The reference to "Beta5" will be stripped out in the final TPA Version 3.1 source code.

6. Software Analysis and Results

a. Analysis: FOR_STUDY Version 1.2 was utilized as the software analysis tool for all preliminary Beta versions of TPA Version 3.1. FOR_STUDY was run in the "default configuration" as specified in the Users Manual and numerous, repetitive runs were made on different modules, subroutines and stand alone segments. Documentation was captured from such runs by C. Scherer and was filed in three large white notebooks. FOR_STUDY was utilized extensively on the TPA code, throughout development and through all the TPA Version 3.1 preliminary Beta versions.

b. Analysis Report: Portions of one of the final analysis reports has been printed and numerous pages have been attached to this document. C. Scherer answered questions of the CNWRA Code Custodian regarding use of FOR_STUDY and made notes on the analysis report attached. A final FOR_STUDY report was performed on each of the final versions of the code. Error and warning messages were helpful in cleaning up the code and in "bug reduction." Rather than supply a total of warning and error messages, reference is made to the large white notebooks of C. Scherer to show the use of FOR_STUDY to improve the quality of the code. The code developer was cognizant of all such testing and corrections being made throughout the development of TPA Version 3.1.

c. Resolution of Comments: The TPA Version 3.1 developer reviewed all of the warning and error messages with C. Scherer and addressed the most critical problem statements. According to the code developer, the warning and error messages from the many FOR_STUDY runs were evaluated and necessary modifications were made to the TPA Version 3.1 source code. Warning messages are viewed as nonfatal and these are not recognized by the SUN compiler. Error messages not recognized by the UNIX FORTRAN compiler remain, according to the code developer, and these are not pertinent to the functional part of the code. A decision to not address several warning messages in the SNLLHS.f routine was made as that had recently been shipped from the NRC. See the "Draft Handout Annotated Outline of Input Data Preparation for TPA Code Version 3.1" dated September 5, 1997 (included in this TPA Version 3.1 folder).



9/5/97

Sitakanta Mohanty
CNWRA Software Developer



9/5/97

Bruce Mabrito
CNWRA Software Custodian

Attachments/

original to: Software Folder
cc: CNWRA Software Developer

installation test

11/100

To: Bruce Mabrito at CNWRA-OS2
CC: Robert Baca at CNWRA-OS2
CC: Ronald Janetzke at CNWRA-SUN
From: Sitakanta Mohanty
Subject: installation test
09-05-97 08:38 PM

Bruce,

I had conducted an installation test of tpa code version 3.1 on bigbend (an ultra 2). They are consistent with what we have observed with the previous version. No discrepancies were noticed. This test run was also compared against R. Janetzke's run on Scratchy1. No discrepancies were observed.

-Sitakanta

13/100

c =====

c
c "This computer code/material was prepared 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. Neither the developer(s) of the code nor any of their
c sponsors 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 privately-
c owned rights."

c
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 the program (including but not limited to loss of
c data or data being rendered inaccurate or losses sustained by third
c parties or a failure of the program to operate with other programs),
c even if you have been advised of the possibility of such damages or
c for any claim by any other party."

c
c =====

c
c by T. McCartin, S. Mohanty, R. Janetzke, R. Rice, A. Lozano

c
c =====

subroutine uzft(ir, isa, mxntime, mxnnucl,
:q!

14/100

```
c calculates flow & transpc of nuclides from repository t
c water table. This is lower portion of unsaturated zone.
c
c ir      = input, integer, realization number currently being studied
c isa     = input, integer, index for subarea currently being studied
c mxntime = input, integer, used to dimension array
c mxnnucl = input, integer, used to dimension array
c ntim    = input, integer, actual number of times
c tim(ntim) = input, double precision, array of times
c nnucl = input, integer, number of nuclides used to dimension arrays
c qm3peryrinsa(ntim) = input, double precision, flow rate [m^3/yr] from
c      unsaturated zone into saturated zone below repository.
c ciperyrinsaintoloweruz(mxntime,nnucl) = input, double precision,
c      nuclide source flow rate [Ci/yr]
c      in saturated zone below repository [Ci/yr] for each of
c      the 43(=inucl) nuclides
c qm3peryrinsafromuz(ntim) = output, double precision, flow rate [m^3/yr]
c      from uz into water table below repository
c ciperyrinsafromuz(mxntime,nnucl) = output, double precision, nuclide
c      release rate from UZ and into SZ
c tavgwtt = output, double precision, total average uz travel time (yr)
cc (rwr 6/9/97 the above line and the argument list modified for tavgwtt)
c
c
c      implicit double precision (a-h,o-z)
```

15/10

```
print *, ' ***>>> error in uzft <<<*** '  
print *, ' istatus .ne. 0 '  
print *, ' istatus = sh( rm nefii.rel ) '  
print *, ' istatus = ', istatus  
stop  
endif
```

```
cc      print *, ' uzft: calling prenefmks '  
cc rwj 7-14-97 Add counter for number of neftran legs to argument  
cc      list.
```

```
      call prenefmks(isa, mxntime, ntim, tim, nnucl, names,  
&                  ciperyrinsaintoloweruz, qm3peryrinsa,  
&                  tavgwtt, neflegs)  
cc rwr 6/9/97 the argument list modified for tavgwtt
```

```
cc if no release, get out quick
```

```
      sum = 0.0d0  
      do i = 1, nnucl  
        sum = sum + ciperyrinsaintoloweruz( ntim, i)  
      enddo  
      if( sum .lt. 1.0d-15 ) then  
        call zero( mxntime*nnucl, ciperyrinsafromuz(1,1) )  
        return  
      endif
```

```
cc rwj 7-14-97 Remove the following test per telecon with T.McCartin  
cc      on 7-11-97.  
cc  
cc rwj 7-2-97 Return early without running NEFTRAN when ground  
cc      water travel time is less than 10 years for UZ.  
cc
```

16/100

c Program Name: TPA - Total-Syst. Performance Assessment Code
c File Name: peakfinder.f
c File Date: 09/04/97
c Release Version: 3.1
c
c Client Name: USNRC
c U. S. Nuclear Regulatory Commission
c NRC Office of Nuclear Material Safety and Safeguards
c Division of Waste Management
c Contract Number: NRC 02-93-005
c
c NRC Contact: Tim McCartin (301) 415-6681
c
c CNWRA Contact: Sitakanta Mohanty (210) 522-5185
c Center for Nuclear Waste Regulatory Analyses
c San Antonio, Texas 78238-5166
c smohanty@swri.edu
c
c Documentation: "Total-System Performance Assessment (TPA)
c Version 3.1 Code: Module Description and User's
c Guide", Center for Nuclear Waste Regulatory
c Analyses, April 30, 1997
c NUREG-Series Designator: N/A

cc

c
c =====

c
c
c D I S C L A I M E R
c

c
c =====

c
c "This computer code/material was prepared 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. Neither the developer(s) of the code nor any of their
c sponsors 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 privately-
c owned rights."

c
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 the program (including but not limited to loss of
c data or data being rendered inaccurate or losses sustained by third
c parties or a failure of the program to operate with other programs),
c even if you have been advised of the possibility of such damages or
c for any claim by any other party."

c
c =====

c
c by S. Mohanty, R. Janetzke, R. Rice
c R. Manteufel (initial version)
c

c=====

subroutine peakfinder(mxntime, mxnnucl,
& ntim, tim, nnucl, names,

17/680

```

&      remperyrwnr,
&      remperyrwswr )
=====
c find and report peak doses (both time and magnitude of)
c the peaks are for each nuclide
c as well as for summed dose contribution for all nuclides
c
c mxntime = input, integer, size to dimension arrays
c mxnnucl = input, integer, size to dimension arrays
c ntim = input, integer, maximum number of times used to dimension arrays
c tim(ntim) = input, double precision, array of times
c nnucl = input, integer, maximum number of nuclides used to dimension arrays
c names(nnucl) = input, character*6, names of nuclides to be tracked
c remperyrwnr[mxntime,nnucl] = input, double precision, array of
c      annual EDE (effective dose equivalent) per nuclide
c remperyrwswr(ntim) = input, double precision, time-dependent annual
c      EDE summed over all nuclides
c
      implicit double precision (a-h,o-z)

      dimension tim(ntim)
      character*6 names(nnucl)
      dimension remperyrwnr(mxntime, mxnnucl)
      dimension remperyrwswr(mxntime)

      character*60 name
      common / peak1 / ikey

      common / peak5 / iownpeakdosev
      common / peak6 / iownpeakdoset
      common / peak7 / ipeakdosev
      common / peak8 / ipeakdoset
      common / peak9 / iownpeakdosenv(43)
      common / peak10 / iownpeakdosent(43)
      common / peak11 / ipeakdosenv(43)
      common / peak12 / ipeakdosent(43)

      external valuesp

      if( (nnucl .le. 0) .or. (nnucl .gt. 43) ) then
        print *, ' ***>>> Error in PEAKFINDER <<<*** '
        print *, ' (nnucl .le. 0) .or. (nnucl .gt. 43)'
        print *, ' nnucl = ', nnucl
        include 'stop.i'
      endif
      if( ntim .le. 0 ) then
        print *, ' ***>>> Error in PEAKFINDER <<<*** '
        print *, ' ntim .le. 0 '
        print *, ' ntim = ', ntim
        include 'stop.i'
      endif

      if (ikey .ne. 39231) then
        call clearchar( 60, name )
        name = 'Magn of Peak Annual Dose [rem/yr]'
        iownpeakdosev = iaddconsmv( name )
        ipeakdosev = imvquery( name )

        call clearchar( 60, name )
        name = 'Time of Peak Annual Dose [yr]'

```

18/100

```

iownpeakdoset = iaddconsmv( name
ipeakdoset = imvquery( name )

do ii = 1, nnuc1
  call clearchar( 60, name )
  name = 'Magn of Peak Annual Dose from ' // names(ii)
& // ' [rem/yr]'
  iownpeakdosenv(ii) = iaddconsmv( name )
  ipeakdosenv(ii) = imvquery( name )
  call clearchar( 60, name )
  name = 'Time of Peak Annual Dose from ' // names(ii) // ' [yr]'
  iownpeakdosent(ii) = iaddconsmv( name )
  ipeakdosent(ii) = imvquery( name )
enddo

ikey = 39231
endif

do k = 1, nnuc1
  tmax = tim(ntim)
  pmax = remperyrqwnr(ntim,k)
  do it = (ntim-1), 1, -1
    if( remperyrqwnr(it,k) .gt. pmax ) then
      tmax = tim(it)
      pmax = remperyrqwnr(it,k)
    endif
  enddo
  call setconsmv( ipeakdosenv(k), iownpeakdosenv(k), pmax )
  call setconsmv( ipeakdosent(k), iownpeakdosent(k), tmax )
enddo

tmax = tim(ntim)
pmax = remperyrqwsr(ntim)
do it = (ntim-1), 1, -1
  if( remperyrqwsr(it) .gt. pmax ) then
    tmax = tim(it)
    pmax = remperyrqwsr(it)
  endif
enddo
call setconsmv( ipeakdosev, iownpeakdosev, pmax )
call setconsmv( ipeakdoset, iownpeakdoset, tmax )

return
end

```


21/100

```
C=====
      subroutine peakfinder( mxntime, mxnnucl,
&      ntim, tim, nnucl, names,
&      remperyrwnr,
&      remperyrwswr )
C=====
c find and report peak doses (both time and magnitude of)
c the peaks are for each nuclide
c as well as for summed dose contribution for all nuclides
c
c mxntime = input, integer, size to dimension arrays
c mxnnucl = input, integer, size to dimension arrays
c ntim = input, integer, maximum number of times used to dimension arrays
c tim(ntim) = input, double precision, array of times
c nnucl = input, integer, maximum number of nuclides used to dimension arrays
c names(nnucl) = input, character*6, names of nuclides to be tracked
c remperyrwnr[mxntime,nnucl] = input, double precision, array of
c      annual EDE (effective dose equivalent) per nuclide
c remperyrwswr(ntim) = input, double precision, time-dependent annual
c      EDE summed over all nuclides
c
      implicit double precision (a-h,o-z)

      dimension tim(ntim)
      character*6 names(nnucl)
      dimension remperyrwnr(mxntime, mxnnucl)
```

```
      real*8 pntlast,tprii
c*** end of declarations inserted by spag
c-----
c  code:      ebspac (engineering barrier system performance assessment code)
c
c  part:      ebspac_fail.f.  this is one of the two parts of ebspac
c
c  version:   1.0
c
c  date:      january 6, 1997
c
c  purpose:   the ebspac computer program has been developed jointly by the
c              center for nuclear waste regulatory analyses (cnwra) and the
c              u. s. nuclear regulatory commission (nrc) for use in the
c              review of doe's licence application for the hlw geologic
c              repository by the office of nmss, dwm.
c
c              ebspac_fail.f part of ebspac calculates waste package failure
c              time & ebspac_release.f part of ebspac calculates the rate of
c              release of nuclides from the repository in the near field.
c
c              the repository is assumed to be made up of cells and ebspac
c              provides results for one cell. input to ebspac is read from
c              files which may either be created manually by the user or
c              generated by use of codes other than ebspac.
c
c  developers: ebspac_fail.f was developed by sitakanta mohanty
c              at the cnwra with assistance from tae ahn (nrc, 301-415-5812),
c              g. cragnolino, p. lichtner, & n. sridhar
c
c  user:      developed for use in nrc iterative performance assessment
c              iii. this computer code is managed under cnwra's code
c              configuration procedure. any modifications to the source
c              code must be reported to code custodian (see below).
c              other versions of this code will be released in the future.
c
c  disclaimer: this computer code has not been formally or informally
c
```

23/00

```
c      prepare and read input data including temperature rel. hum. history
      call input
c AUG28.97 JB - Added close statements to decreaset the number of open files
      close (9)
      close (22)
c AUG28.97 JB End

c      iflagtpa = 2
      print*, 'iflagtpa=', iflagtpa
      nhista = int(tend/timintv)

      call temphstry (iflag, iflagtpa, nset, timr, tcan, tavg, humd,
&          tend, nhist, nhista, timintv)

      if (iflagtpa.eq.1) then

c      the following file must be created by the tpa code using multiflo
c      data. Otherwise run chloride.f program to generate this file
c      using multiflo data.

c      print*, 'nintv,nhist,timr=', nintv, nhist, timr
      call multiflodata(nintv, nhist, timr)
      open (21, file='chloridemf.dat', status='unknown')
csm5/12      call chloride(iflag, iflagtpa, cfactor, nset, timr,
csm5/12      &          clconc, nhist)
      else

      open (21, file='chloride.dat', status='unknown')

c-->
```

24/10

Input file tpa.inp as supplied with TPA , sion 3.1beta5 Code.

Not a basecase dataset. Use only for beta testing.

TPA 3.1beta5, Job started: Thu Sep 4 20:41:58 1997

Subarea Averaged Infiltration/Deep Percolation Including

After Reflux and Diversion - Values for Each Vector

| vector | time | avinfil | avreflux | avdivert |
|----------|------------|------------|------------|------------|
| unitless | yr | mm/yr | mm/yr | mm/yr |
| 1 | 0.0000E+00 | 6.6151E+00 | 2.9015E+00 | 8.7045E-01 |
| 1 | 2.5694E+01 | 6.6151E+00 | 8.3585E+00 | 2.5075E+00 |
| 1 | 5.8078E+01 | 6.6151E+00 | 8.7162E+00 | 2.6149E+00 |
| 1 | 9.8894E+01 | 6.6151E+00 | 8.6459E+00 | 2.5938E+00 |
| 1 | 1.5034E+02 | 6.6151E+00 | 8.5442E+00 | 2.5633E+00 |
| 1 | 2.1518E+02 | 6.6151E+00 | 8.4231E+00 | 2.5269E+00 |
| 1 | 2.9690E+02 | 6.6151E+00 | 8.2811E+00 | 2.4843E+00 |
| 1 | 3.9990E+02 | 6.6151E+00 | 8.1180E+00 | 2.4354E+00 |
| 1 | 5.2972E+02 | 6.1539E+00 | 7.4737E+00 | 2.2421E+00 |
| 1 | 6.9334E+02 | 6.1539E+00 | 7.2744E+00 | 2.1823E+00 |
| 1 | 8.9957E+02 | 6.1539E+00 | 7.0655E+00 | 2.1196E+00 |
| 1 | 1.1595E+03 | 5.7589E+00 | 6.4618E+00 | 1.9385E+00 |
| 1 | 1.4871E+03 | 5.5414E+00 | 6.0481E+00 | 1.8144E+00 |
| 1 | 1.9000E+03 | 5.4257E+00 | 5.7607E+00 | 1.7282E+00 |
| 1 | 2.4204E+03 | 5.1484E+00 | 5.3468E+00 | 1.6041E+00 |
| 1 | 3.0764E+03 | 4.7473E+00 | 4.8506E+00 | 1.4552E+00 |
| 1 | 3.9031E+03 | 4.6172E+00 | 4.6623E+00 | 1.3987E+00 |
| 1 | 4.9451E+03 | 4.4856E+00 | 4.5015E+00 | 1.3505E+00 |
| 1 | 6.2584E+03 | 4.6058E+00 | 4.6102E+00 | 1.3831E+00 |
| 1 | 7.9137E+03 | 5.2684E+00 | 5.2691E+00 | 1.5807E+00 |
| 1 | 1.0000E+04 | 6.5620E+00 | 6.5621E+00 | 1.9686E+00 |

25/100

pcl4

JOB 510

all_exec.rpt

*Objective Evidence
of FOR-STUDY
NEWS. Provided by
C. Scherer 9/5/97*

*all 26 tpa files together
on forstudy run* *BEW*

*Does not include
stand alone's.
BEW*

For: scratchy1!cscherer
Date: Fri Sep 5 08:16:39 CDT 1997
Submit queue: Ethernet
Submitted: 295:20:12
Started: 295:20:12



QMS 1725 Print System

QMS 1725 (1st floor)

Warning #700 array.f,409: No interface for routine D_ERF()
Warning #700 ashplumo.f,85: No interface for routine SH()
Warning #700 exec.f,267: No interface for routine TIME()
Warning #700 exec.f,268: No interface for routine CTIME()

26/100
} these warning messages can be
resolved by adding an
interface for each function to
the .ntf file - no changes
are necessary to the code.

CSS

Note: Most of the warning #700 message contained
in the report from forstudy runs on single .f
files are resolved when forstudy is run on all
the tpa .f files at once.

CSS

Carol S. Scherer

pcl4

JOB 470

array.rpt

*forstudy run on array. f only
- no messages generated*

*26 Files run
individually. be*

NO messages on array

For: cscherer
Date: Fri Sep 5 07:53:38 CDT 1997
Submit queue: Ethernet
Submitted: 294:56:48
Started: 294:56:48

pcl4

JOB 471

ashplumo.rpt

for study analysis on ashplumo, f only

For: cscherer
Date: Fri Sep 5 07:53:51 CDT 1997
Submit queue: Ethernet
Submitted: 294:57:00
Started: 294:57:01

Warning #700 ashplumo.f,83: No interface f. routine CLEARCHAR()
Warning #700 ashplumo.f,85: No interface for routine SH()
Warning #700 ashplumo.f,94: No interface for routine ISPQUERY()
Warning #700 ashplumo.f,191: No interface for routine IADDCONSMV()
Warning #700 ashplumo.f,192: No interface for routine IMVQUERY()
Warning #700 ashplumo.f,198: No interface for routine VALUESP()
Warning #700 ashplumo.f,221: No interface for routine IVALUESP()
Warning #700 ashplumo.f,249: No interface for routine SETCONSMV()
Warning #700 ashplumo.f,330: No interface for routine IGETUNITNUMBER()

can be resolved by adding
interface to .ntt file

29/100

} these
functions
are
contained
in other .f
files

(55)

pcl4

JOB 472

ashrmovo.rpt

for study run on ashrmovo.t only

For: cscherer
Date: Fri Sep 5 07:54:04 CDT 1997
Submit queue: Ethernet
Submitted: 294:57:12
Started: 294:57:13



QMS 1725 Print System

QMS 1725 (1st floor)

Warning #700 ashrmovo.f,172: No interface routine CLEARCHAR()
Warning #700 ashrmovo.f,174: No interface for routine ISPQUERY()
Warning #700 ashrmovo.f,420: No interface for routine VALUESP()
Warning #700 ashrmovo.f,527: No interface for routine NEWINVENTDB()
Warning #700 ashrmovo.f,533: No interface for routine HALFLIFEPERISO()
Warning #700 ashrmovo.f,562: No interface for routine DECAY43MOL()
Warning #700 ashrmovo.f,597: No interface for routine DECAYREMOVE43MOL()

31/1/80
} these functions are
contained in other . + files
CSS

Warning #700 array.f,409: No interface for routine D_ERF()

warning can be resolved by
adding an interface to .intf file.
no changes to array.f are
necessary

css

pcl4

JOB 476

condxyzt.rpt

forstudy run on condxyzt.f only

For: cscherer
Date: Fri Sep 5 07:54:53 CDT 1997
Submit queue: Ethernet
Submitted: 294:57:59
Started: 294:57:59

Warning #700 condxyzt.f,138: No interface . routine QGAUS()
Warning #700 condxyzt.f,201: No interface for routine QPERMTU()
Warning #700 condxyzt.f,216: No interface for routine D_ERF()
Warning #700 condxyzt.f,220: No interface for routine DMYEXP()

add interface to .intf file

34/100
} these
functions
are contained
in other .f
files.
(5)

pcl4

JOB 477

dcags.rpt

forstudy run on dcags.f only

For: cscherer
Date: Fri Sep 5 07:55:06 CDT 1997
Submit queue: Ethernet
Submitted: 294:58:11
Started: 294:58:11

Warning #700 dcags.f,134: No interface for routine CLEARCHAR()
Warning #700 dcags.f,136: No interface for routine ISPQUERY()
Warning #700 dcags.f,150: No interface for routine IGETUNITNUMBER()
Warning #700 dcags.f,160: No interface for routine SH() - add interface to .ntf file
Warning #700 dcags.f,210: No interface for routine VALUESP()
Warning #700 dcags.f,396: No interface for routine INDEXPERISO()

36/100

} these functions
are contained
in other .f
files
c ss

pcl4

JOB 478

dcagw.rpt

forstudy run on dcagw.f only

For: cscherer
Date: Fri Sep 5 07:55:18 CDT 1997
Submit queue: Ethernet
Submitted: 294:58:23
Started: 294:58:23

Warning #700 dcagw.f,172: No interface for routine CLEARCHAR()
Warning #700 dcagw.f,174: No interface for routine ISPQUERY()
Warning #700 dcagw.f,205: No interface for routine IGETUNITNUMBER()
Warning #700 dcagw.f,215: No interface for routine SH() - add interface to .wff
Warning #700 dcagw.f,283: No interface for routine VALUESP()
Warning #700 dcagw.f,518: No interface for routine INDEXPERISO()

38/100
} these functions
are contained
in the .f
files.
CSS

pcl4

JOB 479

ebsfail.rpt

for study run on ebsfail. f only

For: cscherer
Date: Fri Sep 5 07:55:31 CDT 1997
Submit queue: Ethernet
Submitted: 294:58:35
Started: 294:58:35

Warning #700 ebsfail.f,176: No interface for routine IGETUNITNUMBER()
Warning #700 ebsfail.f,184: No interface for routine CLEARCHAR()
Warning #700 ebsfail.f,189: No interface for routine SH() ~ add interface to entry file
Warning #700 ebsfail.f,203: No interface for routine GNSA()
Warning #700 ebsfail.f,208: No interface for routine IADDCONSMV()
Warning #700 ebsfail.f,209: No interface for routine IMVQUERY()
Warning #700 ebsfail.f,216: No interface for routine ISPQUERY()
Warning #700 ebsfail.f,513: No interface for routine VALUESP()
Warning #700 ebsfail.f,573: No interface for routine IVALUESP()
Warning #700 ebsfail.f,964: No interface for routine SETCONSMV()
Warning #700 ebsfail.f,979: No interface for routine MAPTIMEOFEVENT()

} these functions are contained in other .f files.
CSS

4/1/00

pcl4

JOB 480

ebsrel.rpt

for study run on ebsrel.f only

For: cscherer
Date: Fri Sep 5 07:55:44 CDT 1997
Submit queue: Ethernet
Submitted: 294:58:47
Started: 294:58:47

Warning #700 ebsrel.f,218: No interface for routine IGETUNITNUMBER()
Warning #700 ebsrel.f,229: No interface for routine CLEARCHAR()
Warning #700 ebsrel.f,235: No interface for routine SH() - add interface to .ntf file
Warning #700 ebsrel.f,252: No interface for routine ISPQUERY()
Warning #700 ebsrel.f,533: No interface for routine INDEXPERISO()
Warning #700 ebsrel.f,576: No interface for routine GSANWP()
Warning #700 ebsrel.f,583: No interface for routine VALUESF()
Warning #700 ebsrel.f,760: No interface for routine IVALUESF()
Warning #700 ebsrel.f,945: No interface for routine DECAY43MOL()
Warning #700 ebsrel.f,953: No interface for routine UCLJS()
Warning #700 ebsrel.f,955: No interface for routine WMOLEPERISO()
Warning #700 ebsrel.f,956: No interface for routine HALFLIFEPERISO()
Warning #700 ebsrel.f,958: No interface for routine ACTIVITYPERISO()
Warning #700 ebsrel.f,1117: No interface for routine NAMEISO()
Warning #700 ebsrel.f,1119: No interface for routine ISONEOFSET()
Warning #700 ebsrel.f,1123: No interface for routine MAPLIST()
Warning #700 ebsrel.f,1159: No interface for routine GNSA()
Warning #700 ebsrel.f,1169: No interface for routine IADDCONSMV()
Warning #700 ebsrel.f,1170: No interface for routine IMVQUERY()
Warning #700 ebsrel.f,1178: No interface for routine SETCONSMV()

these
functions
are
contained in
other of
files
CSS

pcl4

JOB 509

exec.rpt

for study run on exec.f only

For: cscherer
Date: Fri Sep 5 08:14:36 CDT 1997
Submit queue: Ethernet
Submitted: 295:16:50
Started: 295:16:50

Warning #700 exec.f,267: No interface for routine TIME() } add interface to .ntf file
Warning #700 exec.f,268: No interface for routine CTIME()
Warning #700 exec.f,271: No interface for routine IGETUNITNUMBER()
Warning #700 exec.f,288: No interface for routine READER()
Warning #700 exec.f,297: No interface for routine SH() — add interface to .ntf file
Warning #700 exec.f,302: No interface for routine CLEARCHAR()
Warning #700 exec.f,479: No interface for routine ISPQUERY()
Warning #700 exec.f,480: No interface for routine IVALUESP()
Warning #700 exec.f,482: No interface for routine ISCONSTANT()
Warning #700 exec.f,544: No interface for routine GNSA()
Warning #700 exec.f,1197: No interface for routine VALUESP()
Warning #700 exec.f,1407: No interface for routine ZERO()
Warning #700 exec.f,1413: No interface for routine PRINTTITLESPP()
Warning #700 exec.f,1448: No interface for routine NEWREALIZATION()
Warning #700 exec.f,1457: No interface for routine SETAGE()
Warning #700 exec.f,1552: No interface for routine UZFLOW()
Warning #700 exec.f,1556: No interface for routine GSAREA()
Warning #700 exec.f,1574: No interface for routine GSANWP()
Warning #700 exec.f,1575: No interface for routine SCOPY()
Warning #700 exec.f,1595: No interface for routine NFENV()
Warning #700 exec.f,1676: No interface for routine EBSFAIL()
Warning #700 exec.f,1698: No interface for routine SAMPLEHAZARDCURVE()
Warning #700 exec.f,1734: No interface for routine SEISMO()
Warning #700 exec.f,1816: No interface for routine FAULTO()
Warning #700 exec.f,1899: No interface for routine VOLCANO()
Warning #700 exec.f,2037: No interface for routine EBSREL()
Warning #700 exec.f,2109: No interface for routine SORTQR()
Warning #700 exec.f,2186: No interface for routine UZFT()
Warning #700 exec.f,2355: No interface for routine SZFT()
Warning #700 exec.f,2490: No interface for routine ADDTO()
Warning #700 exec.f,2678: No interface for routine DCAGW()
Warning #700 exec.f,2697: No interface for routine INDEXPERISO()
Warning #700 exec.f,2818: No interface for routine PEAKFINDER()
Warning #700 exec.f,2893: No interface for routine ASHPLUMO()
Warning #700 exec.f,2916: No interface for routine ASHRMOVO()
Warning #700 exec.f,2970: No interface for routine DCAGS()
Warning #700 exec.f,3068: No interface for routine PRINTVALUESSP()
Warning #700 exec.f,3069: No interface for routine PRINTTITLESMP()
Warning #700 exec.f,3070: No interface for routine PRINTVALUESMP()
Warning #700 exec.f,3074: No interface for routine PRINTTIMESVALUE()
Warning #700 exec.f,3258: No interface for routine DECAY43MOL()
Warning #700 exec.f,3264: No interface for routine EPALIMPERISO()
Warning #700 exec.f,3286: No interface for routine ACTIVITYPERISO()
Warning #700 exec.f,3340: No interface for routine SCALE()
Warning #700 exec.f,3653: No interface for routine IADDCONSMV()
Warning #700 exec.f,3654: No interface for routine INVQUERY()
Warning #700 exec.f,3698: No interface for routine SETCONSMV()

these functions
are contained
in other
files

pcl4

JOB 482

faulto.rpt

forstudy run on faulto. f only

For: cscherer
Date: Fri Sep 5 07:56:09 CDT 1997
Submit queue: Ethernet
Submitted: 294:59:11
Started: 294:59:11



QMS 1725 Print System

QMS 1725 (1st floor)

46/100

Warning #700 faulto.f,111: No interface for routine CLEARCHAR()
Warning #700 faulto.f,113: No interface for routine ISPQUERY()
Warning #700 faulto.f,186: No interface for routine ZERO()
Warning #700 faulto.f,187: No interface for routine VALUESP()
Warning #700 faulto.f,265: No interface for routine GSAREA()
Warning #700 faulto.f,266: No interface for routine QLHITSA()
Warning #700 faulto.f,300: No interface for routine MAPTIMEOFEVENT()

} these functions are
contained in other .f files
CSS

pcl4

JOB 483

fileunit.rpt

*fastudy run on fileunit.f only
- no messages generated*

For: cscherer
Date: Fri Sep 5 07:56:21 CDT 1997
Submit queue: Ethernet
Submitted: 294:59:23
Started: 294:59:23



QMS 1725 Print System

QMS 1725 (1st floor)

pcl4

JOB 484

findelev.rpt

for study run on findelev. f only

For: cscherer
Date: Fri Sep 5 07:56:34 CDT 1997
Submit queue: Ethernet
Submitted: 294:59:35
Started: 294:59:35

Warning #700 findelev.f,82: No interface for routine IGETUNITNUMBER()

Warning #700 findelev.f,84: No interface for routine CLEARCHAR()

} these functions are contained
in other .f files
CSS

49/100

pcl4

JOB 486

invent.rpt

for study run on invent. / only

For: cscherer
Date: Fri Sep 5 07:56:59 CDT 1997
Submit queue: Ethernet
Submitted: 294:59:59
Started: 294:59:59

Warning #700 invent.f,241: No interface for routine ZERO()
Warning #700 invent.f,614: No interface for routine ACOPY()

5/1/80
} these functions are contained in
other of files.
CSS

pcl4

JOB 489

mv.rpt

forstudy run on mv. f only

For: cscherer
Date: Fri Sep 5 07:57:51 CDT 1997
Submit queue: Ethernet
Submitted: 295:00:48
Started: 295:00:49



QMS 1725 Print System

QMS 1725 (1st floor)

Warning #700 mv.f,87: No interface for routine ZERO()
Warning #700 mv.f,93: No interface for routine IRANU()

} these functions are contained in
other of files
CS

53/100

54/100

pcl4

JOB 490

nfenv.rpt

for study run on nfenv. f only

For: cscherer
Date: Fri Sep 5 07:58:03 CDT 1997
Submit queue: Ethernet
Submitted: 295:01:00
Started: 295:01:00



QMS 1725 Print System

QMS 1725 (1st floor)

Warning #700 nfenv.f,194: No interface for routine IGETUNITNUMBER()
Warning #700 nfenv.f,199: No interface for routine SH() - add interface to .ntf file
Warning #700 nfenv.f,258: No interface for routine CLEARCHAR()
Warning #700 nfenv.f,260: No interface for routine ISPQUERY()
Warning #700 nfenv.f,261: No interface for routine VALUESF()
Warning #700 nfenv.f,324: No interface for routine ISPQUERYNOSTOP()
Warning #700 nfenv.f,422: No interface for routine IVALUESF()
Warning #700 nfenv.f,453: No interface for routine ZERO()
Warning #700 nfenv.f,461: No interface for routine GSAXYM()
Warning #700 nfenv.f,486: No interface for routine COND3DXYZT()
Warning #700 nfenv.f,589: No interface for routine QPERMTU()
Warning #700 nfenv.f,651: No interface for routine GSANWP()
Warning #700 nfenv.f,653: No interface for routine SCALE()
Warning #700 nfenv.f,867: No interface for routine MAPLIST()
Warning #700 nfenv.f,926: No interface for routine MGET_FROM_NAME()
Warning #700 nfenv.f,928: No interface for routine DGET_FROM_NAME()
Warning #700 nfenv.f,955: No interface for routine GSAREA()
Warning #700 nfenv.f,1108: No interface for routine SCOPY()

55/100

these functions are
contained in other .f
files.

CSS

pcl4

JOB 491

numrecip.rpt

*for study run on numrecip. f only
- no warning/error messages generated*

For: cscherer
Date: Fri Sep 5 07:58:16 CDT 1997
Submit queue: Ethernet
Submitted: 295:01:12
Started: 295:01:12



QMS 1725 Print System

QMS 1725 (1st floor)

57/100

pcl4

JOB 492

peakfinder.rpt

foratuds run on peakfinder-f only

For: cscherer
Date: Fri Sep 5 07:58:28 CDT 1997
Submit queue: Ethernet
Submitted: 295:01:24
Started: 295:01:24



QMS 1725 Print System

QMS 1725 (1st floor)

58/100

Warning #700 peakfinder.f,113: No interface for routine CLEARCHAR()
Warning #700 peakfinder.f,115: No interface for routine IADDCONSMV()
Warning #700 peakfinder.f,116: No interface for routine IMVQUERY()
Warning #700 peakfinder.f,147: No interface for routine SETCONSMV()

} these functions are
contained in other .f files
css

pcl4

JOB 493

ran.rpt

for study run on ran.f only

For: cscherer
Date: Fri Sep 5 07:58:41 CDT 1997
Submit queue: Ethernet
Submitted: 295:01:36
Started: 295:01:36

60/100

Warning #700 ran.f,611: No interface for routine SORTQR()
Warning #700 ran.f,751: No interface for routine D_ERF()
Warning #700 ran.f,892: No interface for routine QGAUS()
Warning #700 ran.f,922: No interface for routine BETA()

- add interface to interface file
these functions are
contained in other .f
files
css

61/100

pcl4

JOB 494

reader.rpt

for study run on reader. f only

For: cscherer
Date: Fri Sep 5 07:58:54 CDT 1997
Submit queue: Ethernet
Submitted: 295:01:48
Started: 295:01:48

62/100

Warning #700 reader.f,189: No interface fo outline NEWSADB()
Warning #700 reader.f,193: No interface for routine CLEARCHAR()
Warning #700 reader.f,237: No interface for routine ADDCONSTANTPDF()
Warning #700 reader.f,242: No interface for routine ADDIUNIFORMPDF()
Warning #700 reader.f,273: No interface for routine ADDUNIFORMPDF()
Warning #700 reader.f,290: No interface for routine ADDIUNIFORMPDF()
Warning #700 reader.f,307: No interface for routine ADDLOGUNIFORMPDF()
Warning #700 reader.f,324: No interface for routine ADDNORMALPDF()
Warning #700 reader.f,341: No interface for routine ADDLOGNORMALPDF()
Warning #700 reader.f,359: No interface for routine ADDTRIANGULARPDF()
Warning #700 reader.f,388: No interface for routine ADDLOGTRIANGULARPDF()
Warning #700 reader.f,418: No interface for routine ADDBETAPDF()
Warning #700 reader.f,459: No interface for routine ADDLOGBETAPDF()
Warning #700 reader.f,475: No interface for routine ADDEXPONENTIALPDF()
Warning #700 reader.f,517: No interface for routine ADDFINITEEXPONENTIALPDF()
Warning #700 reader.f,547: No interface for routine CHECKINORDER()
Warning #700 reader.f,573: No interface for routine ADDHAZARDCURVE()
Warning #700 reader.f,605: No interface for routine ADDCORREL()
Warning #700 reader.f,644: No interface for routine CHECKFORDUPLICATES()
Warning #700 reader.f,656: No interface for routine ADDUSERDIST()
Warning #700 reader.f,659: No interface for routine INDEXPERISO()
Warning #700 reader.f,726: No interface for routine INDEXPERISONOSTOP()
Warning #700 reader.f,762: No interface for routine ISONEOFSET()
Warning #700 reader.f,803: No interface for routine ICHECKFORDUPLICATES()
Warning #700 reader.f,867: No interface for routine QUADRILATERAL()
Warning #700 reader.f,942: No interface for routine ISPQUERYNOSTOP()
Warning #700 reader.f,944: No interface for routine ISCONSTANT()
Warning #700 reader.f,953: No interface for routine VALUESP()
Warning #700 reader.f,962: No interface for routine SETRAN()
Warning #700 reader.f,970: No interface for routine ISPQUERY()
Warning #700 reader.f,1032: No interface for routine SSADB()
Warning #700 reader.f,1050: No interface for routine NEWMVDB()
Warning #700 reader.f,1089: No interface for routine IVALUESP()

these functions are
contained in other
of files
css

pcl4

JOB 495

sampler.rpt

for study run on sampler. f only

For: cscherer
Date: Fri Sep 5 07:59:06 CDT 1997
Submit queue: Ethernet
Submitted: 295:02:00
Started: 295:02:00

64/100

Warning #700 sampler.f,114: No interface for routine ZEROI()
Warning #700 sampler.f,117: No interface for routine ZERO()
Warning #700 sampler.f,127: No interface for routine CLEARCHAR()
Warning #700 sampler.f,886: No interface for routine SH() - add interface to .ntf file
Warning #700 sampler.f,976: No interface for routine IGETUNITNUMBER()
Warning #700 sampler.f,1011: No interface for routine RANU()
Warning #700 sampler.f,1013: No interface for routine RANLU()
Warning #700 sampler.f,1015: No interface for routine RANN()
Warning #700 sampler.f,1017: No interface for routine RANLN()
Warning #700 sampler.f,1019: No interface for routine RANB()
Warning #700 sampler.f,1021: No interface for routine RANLB()
Warning #700 sampler.f,1023: No interface for routine RANT()
Warning #700 sampler.f,1025: No interface for routine RANLT()
Warning #700 sampler.f,1029: No interface for routine IRANU()
Warning #700 sampler.f,1031: No interface for routine RANE()
Warning #700 sampler.f,1033: No interface for routine RANFE()
Warning #700 sampler.f,1036: No interface for routine RANSET()
Warning #700 sampler.f,1170: No interface for routine GASDEV()
Warning #700 sampler.f,1175: No interface for routine CDFN()
Warning #700 sampler.f,1195: No interface for routine QUANTU()
Warning #700 sampler.f,1197: No interface for routine QUANTLU()
Warning #700 sampler.f,1199: No interface for routine QUANTN()
Warning #700 sampler.f,1201: No interface for routine QUANTLN()
Warning #700 sampler.f,1203: No interface for routine QUANTB()
Warning #700 sampler.f,1205: No interface for routine QUANTLB()
Warning #700 sampler.f,1207: No interface for routine QUANTT()
Warning #700 sampler.f,1209: No interface for routine QUANTLT()
Warning #700 sampler.f,1213: No interface for routine IQANTU()
Warning #700 sampler.f,1215: No interface for routine QUANTE()
Warning #700 sampler.f,1217: No interface for routine QUANTFE()
Warning #700 sampler.f,1220: No interface for routine QUANTSET()
Warning #700 sampler.f,1764: No interface for routine CHECKINORDER()
Warning #700 sampler.f,1907: No interface for routine RAN1()
Warning #700 sampler.f,2280: No interface for routine ISONEOFSET()

these functions are
contained in other .f files
css

pcl4

JOB 496

seismo.rpt

for study run on seismo. f only

For: cscherer
Date: Fri Sep 5 07:59:19 CDT 1997
Submit queue: Ethernet
Submitted: 295:02:12
Started: 295:02:12

Warning #700 seismo.f,233: No interface for routine SH()
Warning #700 seismo.f,239: No interface for routine IGETUNITNUMBER()
Warning #700 seismo.f,307: No interface for routine CLEARCHAR()
Warning #700 seismo.f,309: No interface for routine ISPQUERY()
Warning #700 seismo.f,310: No interface for routine VALUESP()
Warning #700 seismo.f,517: No interface for routine IVALUESP()

- add interface - .ntf file

66/100

} these functions are
contained in other .f files
cjs

67/100

pcl4

JOB 497

subarea.rpt

*forstudy run on subarea.f only
- no message generated*

For: cscherer
Date: Fri Sep 5 07:59:31 CDT 1997
Submit queue: Ethernet
Submitted: 295:02:24
Started: 295:02:24

68/100

pcl4

JOB 498

szft.rpt

for study run ~ szft. f nly

For: cscherer
Date: Fri Sep 5 07:59:44 CDT 1997
Submit queue: Ethernet
Submitted: 295:02:36
Started: 295:02:36

69/687

Warning #700 szft.f,154: No interface for routine RUNNEFMKS()
Warning #700 szft.f,156: No interface for routine AFTNEFMKS()
Warning #700 szft.f,159: No interface for routine SH() - add
Warning #700 szft.f,398: No interface for routine VALUESP()
Warning #700 szft.f,398: No interface for routine ISPQUERY()
Warning #700 szft.f,434: No interface for routine GETCHAINS()
Warning #700 szft.f,439: No interface for routine GETELEMENTS()
Warning #700 szft.f,456: No interface for routine INDEXPERISO()
Warning #700 szft.f,457: No interface for routine LJS2()
Warning #700 szft.f,457: No interface for routine NAMEELEM()
Warning #700 szft.f,459: No interface for routine UCLJS()
Warning #700 szft.f,460: No interface for routine LJS()
Warning #700 szft.f,461: No interface for routine WMOLEPERISO()
Warning #700 szft.f,484: No interface for routine HALFLIFEPERISO()
Warning #700 szft.f,485: No interface for routine EPALIMPERISO()
Warning #700 szft.f,591: No interface for routine PUTGWTT()
Warning #700 szft.f,600: No interface for routine IGETUNITNUMBER()
Warning #700 szft.f,605: No interface for routine OPNFIL()
Warning #700 szft.f,890: No interface for routine WRITESOURCE()
Warning #700 szft.f,1000: No interface for routine CLEARCHAR()
Warning #700 szft.f,1201: No interface for routine GSAXY()
Warning #700 szft.f,1308: No interface for routine QLHITSA()
Warning #700 szft.f,1355: No interface for routine QPHITSA()
Warning #700 szft.f,1388: No interface for routine LINEHITLINE()
Warning #700 szft.f,1472: No interface for routine GETVERTLAYERS()
Warning #700 szft.f,1750: No interface for routine TRIANGLE()

interface to interface (.rtf)
file

these functions are
contained in other .f files
c55

pcl4

JOB 499

uzflow.rpt

forestudy run on uzflow. f nly

For: cscherer
Date: Fri Sep 5 07:59:57 CDT 1997
Submit queue: Ethernet
Submitted: 295:02:48
Started: 295:02:48

7/1/80

Warning #700 uzflow.f,101: No interface . routine MAPLIST()
Warning #700 uzflow.f,368: No interface for routine SH()
Warning #700 uzflow.f,398: No interface for routine DGET_FROM_NAME()
Warning #700 uzflow.f,403: No interface for routine MGET_FROM_NAME()
Warning #700 uzflow.f,509: No interface for routine VALUESP()
Warning #700 uzflow.f,514: No interface for routine IVALUESP()
Warning #700 uzflow.f,711: No interface for routine ZERO()
Warning #700 uzflow.f,911: No interface for routine ZEROI()
Warning #700 uzflow.f,926: No interface for routine GSAXY()
Warning #700 uzflow.f,982: No interface for routine QPHITSA()
Warning #700 uzflow.f,1106: No interface for routine IGETUNITNUMBER()
Warning #700 uzflow.f,1156: No interface for routine CLEARCHAR()
Warning #700 uzflow.f,1165: No interface for routine GNSA()
Warning #700 uzflow.f,1171: No interface for routine ACOPY()
Warning #700 uzflow.f,1584: No interface for routine GSAREA()

- add interface to .ntf file

these functions are
contained in other .f files
CS

pcl4

JOB 500

uzft.rpt

foratidy run on uzft.f only

For: cscherer
Date: Fri Sep 5 08:00:09 CDT 1997
Submit queue: Ethernet
Submitted: 295:03:00
Started: 295:03:00



QMS 1725 Print System

QMS 1725 (1st floor)

Warning #700 uzft.f,146: No interface for routine CLEARCHAR()
Warning #700 uzft.f,148: No interface for routine ISPQUERY()
Warning #700 uzft.f,162: No interface for routine IVALUESP()
Warning #700 uzft.f,170: No interface for routine SCOPY()
Warning #700 uzft.f,176: No interface for routine SH() - add interface to .ntf file
Warning #700 uzft.f,219: No interface for routine ZERO()
Warning #700 uzft.f,243: No interface for routine ACOPIY()
Warning #700 uzft.f,877: No interface for routine VALUESP()
Warning #700 uzft.f,1032: No interface for routine INDEXPERISO()
Warning #700 uzft.f,1033: No interface for routine NAMEELEM()
Warning #700 uzft.f,1037: No interface for routine WMOLEPERISO()
Warning #700 uzft.f,1061: No interface for routine HALFLIFEPERISO()
Warning #700 uzft.f,1062: No interface for routine EPALIMPERISO()
Warning #700 uzft.f,1304: No interface for routine GSAREA()
Warning #700 uzft.f,1323: No interface for routine MAPLIST()
Warning #700 uzft.f,1709: No interface for routine IGETUNITNUMBER()
Warning #700 uzft.f,3099: No interface for routine GNSA()
Warning #700 uzft.f,3112: No interface for routine IADDCONSMV()
Warning #700 uzft.f,3113: No interface for routine INVQUERY()
Warning #700 uzft.f,3140: No interface for routine SETCONSMV()

These functions are
contained in other files
css

pcl4

JOB 501

volcano.rpt

for study run on volcano only

For: cscherer
Date: Fri Sep 5 08:00:22 CDT 1997
Submit queue: Ethernet
Submitted: 295:03:12
Started: 295:03:12



QMS 1725 Print System

QMS 1725 (1st floor)

Warning #700 volcano.f,116: No interface routine CLEARCHAR()
Warning #700 volcano.f,118: No interface for routine ISPQUERY()
Warning #700 volcano.f,151: No interface for routine VALUESP()
Warning #700 volcano.f,159: No interface for routine IADDCONSMV()
Warning #700 volcano.f,160: No interface for routine IMVQUERY()
Warning #700 volcano.f,215: No interface for routine ZERO()
Warning #700 volcano.f,219: No interface for routine GSAMTU()
Warning #700 volcano.f,220: No interface for routine GSAREA()
Warning #700 volcano.f,221: No interface for routine QLHITSA()
Warning #700 volcano.f,241: No interface for routine QCHITSA()
Warning #700 volcano.f,265: No interface for routine MAPTIMEOFEVENT()
Warning #700 volcano.f,281: No interface for routine SETCONSMV()

*these functions are
contained in other .f file
css*

pcl4

JOB 502

ashplume.rpt

*forstudy run in ... /codes/
ashplume.f
- no messages generated*

For: cscherer
Date: Fri Sep 5 08:00:56 CDT 1997
Submit queue: Ethernet
Submitted: 295:03:34
Started: 295:03:34

27/100

pcl4

JOB 503

failt.rpt

for study run on ... / codes / failt. f

For: cscherer
Date: Fri Sep 5 08:00:58 CDT 1997
Submit queue: Ethernet
Submitted: 295:03:46
Started: 295:03:46

Warning #700 fault.f,175: No interface f outline FDATE() - can be resolved , adding interface to -utl file
Warning #686 fault.f,875: different type and size used in arg #12 of RKQC()

78/100

this message references code which is never used due to other code which calls it being commented out - does not affect code performance
cs,

79/100

pcl4

JOB 504

nefmks.rpt

for study run on /code/ nefmks.f

For: cscherer
Date: Fri Sep 5 08:01:10 CDT 1997
Submit queue: Ethernet
Submitted: 295:03:58
Started: 295:03:58



QMS 1725 Print System

QMS 1725 (1st floor)

Warning #700 nefmks.f,456: No interface

routine FDATE()

— can be resolved by adding
interface to .ulf file
80/100
css

8/10

pcl4

JOB 505

reaset.rpt

for study run on .../codes/reaset.f

For: cscherer
Date: Fri Sep 5 08:01:23 CDT 1997
Submit queue: Ethernet
Submitted: 295:04:10
Started: 295:04:10

Warning #700 releaset.f,350: No interf

or routine FDATE()

— can be solved by adding
interface to .ulf file

82/100

pcl4

JOB 507

/

snllhs.rpt.orig

*forstudy run on .../code/snllhs.f
L w/o any forstudy
mody*

For: cscherer
Date: Fri Sep 5 08:04:24 CDT 1997
Submit queue: Ethernet
Submitted: 295:06:56
Started: 295:06:56

84/1150

Warning #700 snllhs.f,94: No interface for routine IEEE_HANDLER()
Warning #700 snllhs.f,404: No interface for routine EXIT()
Warning #72 snllhs.f,429: common /WORKX/ was not used in function LHS()
Warning #72 snllhs.f,491: common /SAMP/ was not used in function BANNER()
Warning #72 snllhs.f,597: common /SAMP/ was not used in function CHKDIM()
Warning #72 snllhs.f,677: common /PARAM/ was not used in function CHKSTR()
Warning #72 snllhs.f,677: common /SAMP/ was not used in function CHKSTR()
Warning #72 snllhs.f,849: common /SAMP/ was not used in function CHLSKY()
Warning #543 snllhs.f,905: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #72 snllhs.f,947: common /SAMP/ was not used in function CMCRD()
Warning #72 snllhs.f,996: common /SAMP/ was not used in function CONSTA()
Warning #72 snllhs.f,1598: common /SAMP/ was not used in function HISTO()
Warning #72 snllhs.f,1664: common /SAMP/ was not used in function HPSRT()
Error #171 snllhs.f,2165: Local name OUTPUT was referenced but never set
Warning #72 snllhs.f,2451: common /SAMP/ was not used in function MATINV()
Warning #543 snllhs.f,2518: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #626 snllhs.f,2567: differing number of MATINV() arguments; intrf. vs. call
Warning #72 snllhs.f,2746: common /SAMP/ was not used in function OUTCRD()
Warning #72 snllhs.f,3007: common /SAMP/ was not used in function PMTRX()
Warning #72 snllhs.f,3076: common /PARAM/ was not used in function POSDEF()
Warning #72 snllhs.f,3076: common /SAMP/ was not used in function POSDEF()
Warning #719 snllhs.f,3298: expecting array (to set) - string constant passed in arg #1 of XERROR()
Warning #684 snllhs.f,3298: mixed strings and arithmetic objects in arg #1 of XERROR()
Warning #713 snllhs.f,3298: expecting lvalue as scalar argument - constant/expr passed in arg #2 of XERROR()
Warning #713 snllhs.f,3298: expecting lvalue as scalar argument - constant/expr passed in arg #3 of XERROR()
Warning #713 snllhs.f,3298: expecting lvalue as scalar argument - constant/expr passed in arg #4 of XERROR()
Warning #72 snllhs.f,3393: common /SAMP/ was not used in function RANKER()
Warning #436 snllhs.f,3473: parameter PC char constant definition will be truncated
Error #171 snllhs.f,3677: Local name CDUM1 was referenced but never set
Error #171 snllhs.f,3677: Local name CDUM2 was referenced but never set
Warning #72 snllhs.f,3677: common /SAMP/ was not used in function RDPAR()
Warning #103 snllhs.f,3865: expecting REAL constant in DATA - not CHARACTER
Warning #72 snllhs.f,3889: common /SAMP/ was not used in function SETDEF()
Warning #719 snllhs.f,3999: expecting array (to set) - string constant passed in arg #1 of XERROR()
Warning #684 snllhs.f,3999: mixed strings and arithmetic objects in arg #1 of XERROR()
Warning #713 snllhs.f,3999: expecting lvalue as scalar argument - constant/expr passed in arg #2 of XERROR()
Warning #713 snllhs.f,3999: expecting lvalue as scalar argument - constant/expr passed in arg #3 of XERROR()
Warning #713 snllhs.f,3999: expecting lvalue as scalar argument - constant/expr passed in arg #4 of XERROR()
Warning #719 snllhs.f,4001: expecting array (to set) - string constant passed in arg #1 of XERROR()
Warning #684 snllhs.f,4001: mixed strings and arithmetic objects in arg #1 of XERROR()
Warning #713 snllhs.f,4001: expecting lvalue as scalar argument - constant/expr passed in arg #2 of XERROR()
Warning #713 snllhs.f,4001: expecting lvalue as scalar argument - constant/expr passed in arg #3 of XERROR()
Warning #713 snllhs.f,4001: expecting lvalue as scalar argument - constant/expr passed in arg #4 of XERROR()
Warning #72 snllhs.f,4690: common /SAMP/ was not used in function VIF()
Warning #72 snllhs.f,4740: common /SAMP/ was not used in function WRTCRD()
Warning #72 snllhs.f,4857: common /SAMP/ was not used in function WRTPAR()
Warning #684 snllhs.f,5138: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5139: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5141: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5143: mixed strings and arithmetic objects in arg #1 of XERSAV()
Warning #684 snllhs.f,5144: mixed strings and arithmetic objects in arg #1 of XERABT()
Warning #684 snllhs.f,5169: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5172: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5173: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5175: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5176: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5209: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5211: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5213: mixed strings and arithmetic objects in arg #1 of XERSAV()
Warning #543 snllhs.f,5615: loss of precision in assignment: REAL*8 to REAL*4

Run of For-Study
on SNLLhs.Rpt. orig
First run after
changes - All but
Two were
does not affect processing Resolved
by the last
run. Ben
9/5/97

} does not affect processing

All but 3 are
warnings. Each has
been addressed and
does not affect
the output. Ben
9/5/97

css

Warning #543 snllhs.f,5617: loss of prec on in assignment: REAL*8 to REAL*4
Warning #70 snllhs.f,5702: common /EVAL/ changed relative to prior use
Warning #71 snllhs.f,5702: common /EVAL/ changed relative to first usage
Warning #71 snllhs.f,5727: common /EVAL/ changed relative to first usage

85/000

these messages resulted when forstudy was run on version of snllhs that contains no forstudy modifications.

all but 1st two messages were successfully resolved (see report #2); however, that version of snllhs does not run successfully with number of observation greater than 1

modified version of snllhs not to be shipped until runtime error (probably due to modification done to resolve forstudy message) is identified & corrected

CSG

pcl4

JOB 506

snllhs.rpt #2

forstudy run on .../codes/snllhs.f
L4 all ~~forstudy~~ *mod*

For: cscherer
Date: Fri Sep 5 08:01:36 CDT 1997
Submit queue: Ethernet
Submitted: 295:04:22
Started: 295:04:22

Warning #700 snllhs.f,99: No interface to routine IEEE_HANDLER()
Warning #700 snllhs.f,409: No interface for routine EXIT()

} cannot escape there by any
method tried
CSS

87/100

TPA V 3.1.1


SOFTWARE RELEASE NOTICE

| | | |
|--|----------------------|--------------------------|
| 1. SRN Number: PA-SRN-206 | | |
| 2. Project Title: TSPA & Technical Integration Code | | Project No. 20-01402-762 |
| 3. SRN Title: TPA Version 3.1.1 Rg 10-4-99 | | |
| 4. Originator/Requestor: Ron Janetzke | | Date: Oct. 1, 1999 |
| 5. Summary of Actions <ul style="list-style-type: none"> <input type="checkbox"/> Release of new software <input type="checkbox"/> Release of modified software: <ul style="list-style-type: none"> <input type="checkbox"/> Enhancements made <input type="checkbox"/> Corrections made <input type="checkbox"/> Change of access software <input checked="" type="checkbox"/> Software Retirement | | |
| 6. Persons Authorized Access | | |
| Name | Read Only/Read-Write | Addition/Change/Delete |
| Ron Janetzke Sitakanta Mohanty Tim McCartin (NRC) M. Rose Byrne (NRC) | RW RW RW RW | |
| 7. Element Manager Approval: <i>Arden Willmeyer</i> | | Date: 10/4/99 |
| 8. Remarks: Version superceded. | | |


SOFTWARE RELEASE NOTICE

| | | |
|--|--|-----------------------------|
| 01. SRN Number: PA-SRN-160 | | |
| 02. Project Title: TSPA & Technical Integration Code | | Project No.: 20-5708-762 |
| 03. SRN Title: TPA Version 3.1.1 | | |
| 04. Originator/Requestor: Bruce Mabrito | | Date: 10/1/97 |
| 05. Summary of Actions <ul style="list-style-type: none"> <input type="checkbox"/> Release of new software <input checked="" type="checkbox"/> Release of modified software: <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Enhancements made <input type="checkbox"/> Corrections made <input type="checkbox"/> Change of access software <input type="checkbox"/> Software Retirement | | |
| 06. Persons Authorized Access | | |
| Name | RO/RW | A/C/D |
| Sitakanta Mohanty Ron Janetzke Tim McCartin Dennis Vinson M. Rose Byrne | READ/WRITE READ/WRITE READ/WRITE READ/WRITE | ADDED DELETE |
| 07. Element Manager Approval: <i>Sitakanta Mohanty for R. Baca</i> | | Date: 10/1/97 |
| 08. Remarks: TPA Version 3.1.1 software was sent to NRC Headquarters on 10/1/97. | | |

SOFTWARE SUMMARY FORM

| | | | |
|--|---|---|--|
| 01. Summary Date: 10/1/97 | 02. Summary prepared by (Name and phone) B. Mabrito (210)522-5149 | 03. Summary Action: Modified | |
| 04. Software Date: 10/1/97 | 05. Short Title: TPA Version 3.1.1 | | |
| 06. Software Title: TPA — Total-System Performance Assessment Computer Code, Version 3.1.1 | | 07. Internal Software ID: None | |
| 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 type="checkbox"/> Scientific/Engineering <input type="checkbox"/> Auxiliary Analyses <input checked="" type="checkbox"/> Total System PA <input type="checkbox"/> Subsystem PA <input type="checkbox"/> Other B. Specific: | |
| 11. Submitting Organization and Address: CNWRA/SwRI 6220 Culebra Road San Antonio, TX 78238 | | 12. Technical Contact(s) and Phone: Sitakanta Mohanty (210)522-5185 | |
| 13. Narrative: The TPA Code consists of the following modules: UZFLOW, NFENV, EBSREL, UZFT, SZFT, DCAGW, FAULTO, SEISMO, VOLCANO, ASHPLUMO, ASHRMVO, DCAGS, LHS EXEC. | | | |
| 14. Computer Platform: SUN Workstation | 15. Computer Operating System: UNIX | 16. Programming Language(s): FORTRAN | 17. Number of Source Program Statements: approx. 30,000 lines |
| 18. Computer Memory Requirements: Unknown | 19. Tape Drives: None | 20. Disk/Drum Units: N/A | 21. Graphics: N/A |
| 22. Other Operational Requierments: None | | | |
| 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 | |
| Software Custodian:  | | Date: 10/1/97 | |

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

Date: October 1, 1997
To: Tim McCartin
From: Sitakanta Mohanty 
Subject: Delivery of TPA code Version 3.1.1

Enclosed please find a copy of TPA code Version 3.1.1 in an 8-mm tape. TPA code Version 3.1.1 is an updated version of TPA code Version 3.1 and includes the following changes suggested by the NRC staff:

- Modifications to the stand-alone code nefmks.f provided by R. Codell to make the code single precision for reducing memory requirement were incorporated. This will amount to a slight change in the accuracy of output data. No changes to CCDF output values were observed.
- Array out-of-bound problem in nefmks.f was fixed. Please note that the copy of the nefmks.f sent to the CNWRA by R. Codell already had necessary changes. These changes were retained to avoid duplication of effort.
- Array dimension for number of subareas was reduced from 100 to 30. Although a value of 7 instead of 30 was suggested by R. Codell, the value of 30 was used to avoid assigning a new version number when CNWRA staff would need to use the code for more than 7 subareas. Based on our teleconference on September 29, R. Codell will change the value to 7 in the NRC copy of TPA 3.1.1 so that the computer memory issue at the NRC can be addressed. We recommend that you assign a version number 3.1.1a following this change so that the next version that may be sent to the NRC will carry 3.1.2.
- The waste package failure data are now reported in the wpsfail.res file for all vectors even if there are zero failures for some vectors. This change was requested by V. Colten-Breadley.
- Changes to releaset.f proposed by R. Codell to speed up the TPA code were not implemented in the version. This was okayed by R. Codell over a telephone conversation.
- Input data set was changed based on data from the NRC staff. These changes included Alluvium Kds and fowfactor. Please note that a new tpa.inp.meanvalues file was created based on the new data. Also, in the tpa.inp.meanvalues data set, only the selected flow parameters provided by you are sampled while other parameters are kept fixed at mean values.

cc: Keith McConnell, R. Baca, B. Sagar, B. Mabrito, R. Janetzke

To: Bruce Mabrito at CNWRA-OS2
CC: Connie Cruz at CNWRA-OS2
CC: Robert Baca at CNWRA-OS2
CC: Henry Garcia at CNWRA-OS2
From: Robin K Juhl
Subject: TPA version 3.1.1
10-02-97 01:06 PM

Sir,

I placed the TPA, version 3.1.1, files from the tape into the Revision Control System directory of the server mammoth (in /lan/rcs/tpa_v3.1.1). The contents of all files, including those in the subdirectories "Release" and "Verify," were placed under SCCS control.

Formal Details --

Program Name: TPA, V3.1.1
Date Entered: 1 Oct 97
Control Method: SCCS
Location: mammoth:/lan/rcs/tpa_v3.1.1

V/R,
Robin Juhl

6/45

SOFTWARE PROBLEM/CHANGE REPORT

| | | |
|--|----------------------------------|--------------------------------|
| 01. SPCR Number: PA-SPCR-203 | | |
| 02. Project Title: TPA3.1 | | 20-5708-762 Project Number: |
| 03. SPCR Title: Comprehensive WPSFAIL.RES Output. | | |
| 04. AFFECTED SOFTWARE AND/OR DOCUMENTATION | | |
| exec.f | | |
| 05. DESCRIPTION OF PROBLEM/CHANGE | | |
| <p>a) File wpsfail.res does not contain output for realizations with \emptyset non-initial canister failure.</p> <p>Also:</p> <p>b) File gwpkdos.res displays year \emptyset for time of drinking water peak dose when peak dose is \emptyset.</p> | | |
| 06. PROBLEM SOLUTION/NEED FOR CHANGE | | |
| <p>a) At least one entry should be contained in wpsfail.res for each realization. If there are no canister failures for a given realization, zeroes should be displayed for time of failure equal to the simulation time.</p> <p>b) The time of drinking water peak dose should be the simulation end time, when the d.w. peak dose is zero.</p> | | |
| 07. Originator: V. Colten-Brady Title: NRC | | Date: 9-15-97 |
| PROJECT | | |
| 08. Need by Date: | 09. Approved: [Signature] | Disapproved: (10/1/97) |
| 10. Software Developer: S. Mohanty et al. | | Date: |
| 11. Element Manager: S. Mohanty for R. Baca | | Date: 10/1/92 |
| 12. IMPLEMENTED SOLUTION | | |
| <p>a) a special loop was added to exec.f after the write to wpsfail.res, for each realization to detect if any event failure times were written. If not, an event time equal to the simulation time is written</p> | | |
| 13. Implemented By: Ron Jones | | Date: 9-22-97 |

followed by all zeroes.

b) The time of drinking water peak dose is initialized to the simulation time before the data is scanned for a peak.

7/45

tpa_v3.1.1/:

total 2430

| | | | | | | | | |
|------------|---|------|-------|--------|-----|---|-------|--------------|
| -r--r--r-- | 1 | root | other | 1241 | Oct | 1 | 17:10 | CLEANUP |
| -r--r--r-- | 1 | root | other | 689 | Oct | 1 | 17:10 | Makefile |
| -r--r--r-- | 1 | root | other | 4190 | Oct | 1 | 17:10 | README |
| drwxr-xr-x | 2 | root | other | 2560 | Oct | 1 | 17:10 | SCCS |
| -r--r--r-- | 1 | root | other | 28754 | Oct | 1 | 17:10 | array.f |
| -r--r--r-- | 1 | root | other | 17906 | Oct | 1 | 17:10 | ashplumo.f |
| -r--r--r-- | 1 | root | other | 23853 | Oct | 1 | 17:10 | ashrmovo.f |
| drwxr-xr-x | 3 | root | other | 512 | Oct | 1 | 17:09 | ccdf |
| drwxr-xr-x | 3 | root | other | 512 | Oct | 1 | 17:07 | codes |
| -r--r--r-- | 1 | root | other | 9746 | Oct | 1 | 17:10 | condxyz.f |
| drwxr-xr-x | 3 | root | other | 1024 | Oct | 1 | 17:08 | data |
| -r--r--r-- | 1 | root | other | 15257 | Oct | 1 | 17:10 | dcags.f |
| -r--r--r-- | 1 | root | other | 21646 | Oct | 1 | 17:10 | dcagw.f |
| -r--r--r-- | 1 | root | other | 31175 | Oct | 1 | 17:10 | ebsfail.f |
| -r--r--r-- | 1 | root | other | 40379 | Oct | 1 | 17:10 | ebsrel.f |
| -r--r--r-- | 1 | root | other | 198250 | Oct | 1 | 17:10 | exec.f |
| -r--r--r-- | 1 | root | other | 1495 | Oct | 1 | 17:10 | execa.i |
| -r--r--r-- | 1 | root | other | 486 | Oct | 1 | 17:10 | execb.i |
| -r--r--r-- | 1 | root | other | 10907 | Oct | 1 | 17:10 | faulto.f |
| -r--r--r-- | 1 | root | other | 5942 | Oct | 1 | 17:10 | fileunit.f |
| -r--r--r-- | 1 | root | other | 5825 | Oct | 1 | 17:10 | findelev.f |
| -r--r--r-- | 1 | root | other | 60 | Oct | 1 | 17:10 | ful.i |
| -r--r--r-- | 1 | root | other | 430 | Oct | 1 | 17:10 | fu2.i |
| -r--r--r-- | 1 | root | other | 55065 | Oct | 1 | 17:10 | invent.f |
| -r--r--r-- | 1 | root | other | 57 | Oct | 1 | 17:10 | inventa.i |
| -r--r--r-- | 1 | root | other | 109 | Oct | 1 | 17:10 | inventb.i |
| -r--r--r-- | 1 | root | other | 194 | Oct | 1 | 17:10 | inventc.i |
| -r--r--r-- | 1 | root | other | 62 | Oct | 1 | 17:10 | inventd.i |
| -r--r--r-- | 1 | root | other | 66 | Oct | 1 | 17:10 | invente.i |
| -r--r--r-- | 1 | root | other | 65 | Oct | 1 | 17:10 | inventf.i |
| -r--r--r-- | 1 | root | other | 64 | Oct | 1 | 17:10 | inventg.i |
| -r--r--r-- | 1 | root | other | 64 | Oct | 1 | 17:10 | inventh.i |
| -r--r--r-- | 1 | root | other | 75 | Oct | 1 | 17:10 | inventi.i |
| -r--r--r-- | 1 | root | other | 72 | Oct | 1 | 17:10 | inventj.i |
| -r--r--r-- | 1 | root | other | 106 | Oct | 1 | 17:10 | inventk.i |
| -r--r--r-- | 1 | root | other | 75 | Oct | 1 | 17:10 | inventl.i |
| -r--r--r-- | 1 | root | other | 78 | Oct | 1 | 17:10 | inventm.i |
| -r--r--r-- | 1 | root | other | 104 | Oct | 1 | 17:10 | inventn.i |
| -r--r--r-- | 1 | root | other | 78 | Oct | 1 | 17:10 | max500yr.i |
| -r--r--r-- | 1 | root | other | 99 | Oct | 1 | 17:10 | maxchain.i |
| -r--r--r-- | 1 | root | other | 186 | Oct | 1 | 17:10 | maxnnucl.i |
| -r--r--r-- | 1 | root | other | 230 | Oct | 1 | 17:10 | maxnsuba.i |
| -r--r--r-- | 1 | root | other | 130 | Oct | 1 | 17:10 | maxntime.i |
| -r--r--r-- | 1 | root | other | 11776 | Oct | 1 | 17:10 | mv.f |
| -r--r--r-- | 1 | root | other | 111 | Oct | 1 | 17:10 | mva.i |
| -r--r--r-- | 1 | root | other | 56 | Oct | 1 | 17:10 | mvb.i |
| -r--r--r-- | 1 | root | other | 57 | Oct | 1 | 17:10 | mvc.i |
| -r--r--r-- | 1 | root | other | 101 | Oct | 1 | 17:10 | mvd.i |
| -r--r--r-- | 1 | root | other | 72 | Oct | 1 | 17:10 | mve.i |
| -r--r--r-- | 1 | root | other | 72 | Oct | 1 | 17:10 | mvf.i |
| -r--r--r-- | 1 | root | other | 52371 | Oct | 1 | 17:10 | nfenv.f |
| -r--r--r-- | 1 | root | other | 94 | Oct | 1 | 17:10 | nintv.i |
| -r--r--r-- | 1 | root | other | 1502 | Oct | 1 | 17:10 | notice.i |
| -r--r--r-- | 1 | root | other | 4810 | Oct | 1 | 17:10 | numrecip.f |
| -r--r--r-- | 1 | root | other | 259 | Oct | 1 | 17:10 | path.i |
| -r--r--r-- | 1 | root | other | 6316 | Oct | 1 | 17:10 | peakfinder.f |
| -r--r--r-- | 1 | root | other | 44387 | Oct | 1 | 17:10 | ran.f |
| -r--r--r-- | 1 | root | other | 48866 | Oct | 1 | 17:10 | reader.f |
| -r--r--r-- | 1 | root | other | 58 | Oct | 1 | 17:10 | reflux2.i |
| -r--r--r-- | 1 | root | other | 84641 | Oct | 1 | 17:10 | sampler.f |
| -r--r--r-- | 1 | root | other | 62 | Oct | 1 | 17:10 | sampler0.i |
| -r--r--r-- | 1 | root | other | 79 | Oct | 1 | 17:10 | sampler1.i |
| -r--r--r-- | 1 | root | other | 62 | Oct | 1 | 17:10 | sampler2.i |
| -r--r--r-- | 1 | root | other | 62 | Oct | 1 | 17:10 | sampler.a.i |

TPA Version

3.1.1

CONTENTS.

Ben

| | | | | | | | | |
|------------|---|------|-------|--------|-----|---|-------|--------------------|
| -r--r--r-- | 1 | root | other | 62 | Oct | 1 | 17:10 | sampler.i |
| -r--r--r-- | 1 | root | other | 62 | Oct | 1 | 17:10 | samplerc.i |
| -r--r--r-- | 1 | root | other | 68 | Oct | 1 | 17:10 | samplerd.i |
| -r--r--r-- | 1 | root | other | 133 | Oct | 1 | 17:10 | samlpere.i |
| -r--r--r-- | 1 | root | other | 111 | Oct | 1 | 17:10 | samplerf.i |
| -r--r--r-- | 1 | root | other | 84 | Oct | 1 | 17:10 | samplerg.i |
| -r--r--r-- | 1 | root | other | 68 | Oct | 1 | 17:10 | samplerh.i |
| -r--r--r-- | 1 | root | other | 83 | Oct | 1 | 17:10 | sampleri.i |
| -r--r--r-- | 1 | root | other | 61 | Oct | 1 | 17:10 | samplerj.i |
| -r--r--r-- | 1 | root | other | 208 | Oct | 1 | 17:10 | samplerk.i |
| -r--r--r-- | 1 | root | other | 104 | Oct | 1 | 17:10 | samplerl.i |
| -r--r--r-- | 1 | root | other | 63 | Oct | 1 | 17:10 | samplerm.i |
| -r--r--r-- | 1 | root | other | 79 | Oct | 1 | 17:10 | samlpern.i |
| -r--r--r-- | 1 | root | other | 63 | Oct | 1 | 17:10 | samlpero.i |
| -r--r--r-- | 1 | root | other | 120 | Oct | 1 | 17:10 | samplerp.i |
| -r--r--r-- | 1 | root | other | 103 | Oct | 1 | 17:10 | samplerq.i |
| -r--r--r-- | 1 | root | other | 176 | Oct | 1 | 17:10 | samlerrr.i |
| -r--r--r-- | 1 | root | other | 224 | Oct | 1 | 17:10 | samplers.i |
| -r--r--r-- | 1 | root | other | 70 | Oct | 1 | 17:10 | samlpert.i |
| -r--r--r-- | 1 | root | other | 69 | Oct | 1 | 17:10 | sampleru.i |
| -r--r--r-- | 1 | root | other | 62 | Oct | 1 | 17:10 | samlperv.i |
| -r--r--r-- | 1 | root | other | 60 | Oct | 1 | 17:10 | samplerw.i |
| -r--r--r-- | 1 | root | other | 90 | Oct | 1 | 17:10 | samlperx.i |
| -r--r--r-- | 1 | root | other | 60 | Oct | 1 | 17:10 | samlpery.i |
| -r--r--r-- | 1 | root | other | 60 | Oct | 1 | 17:10 | samlperz.i |
| -r--r--r-- | 1 | root | other | 38504 | Oct | 1 | 17:10 | seismo.f |
| -r--r--r-- | 1 | root | other | 1708 | Oct | 1 | 17:10 | setfiles.i |
| -r--r--r-- | 1 | root | other | 144 | Oct | 1 | 17:10 | stop.i |
| -r--r--r-- | 1 | root | other | 35111 | Oct | 1 | 17:10 | subarea.f |
| -r--r--r-- | 1 | root | other | 134 | Oct | 1 | 17:10 | subareaa.i |
| -r--r--r-- | 1 | root | other | 79 | Oct | 1 | 17:10 | subareab.i |
| -r--r--r-- | 1 | root | other | 82 | Oct | 1 | 17:10 | subareac.i |
| -r--r--r-- | 1 | root | other | 81 | Oct | 1 | 17:10 | subaread.i |
| -r--r--r-- | 1 | root | other | 77 | Oct | 1 | 17:10 | subareae.i |
| -r--r--r-- | 1 | root | other | 57739 | Oct | 1 | 17:10 | szft.f |
| -r--r--r-- | 1 | root | other | 60 | Oct | 1 | 17:10 | szft.i |
| -r--r--r-- | 1 | root | other | 43378 | Oct | 1 | 17:10 | tpa.inp |
| -r--r--r-- | 1 | root | other | 44967 | Oct | 1 | 17:10 | tpa.inp.meanvalues |
| -r--r--r-- | 1 | root | other | 314 | Oct | 1 | 17:10 | uz_climi.i |
| -r--r--r-- | 1 | root | other | 926 | Oct | 1 | 17:10 | uz_climr.i |
| -r--r--r-- | 1 | root | other | 341 | Oct | 1 | 17:10 | uz_climz.i |
| -r--r--r-- | 1 | root | other | 427 | Oct | 1 | 17:10 | uz_flowi.i |
| -r--r--r-- | 1 | root | other | 881 | Oct | 1 | 17:10 | uz_flowr.i |
| -r--r--r-- | 1 | root | other | 176 | Oct | 1 | 17:10 | uz_flowz.i |
| -r--r--r-- | 1 | root | other | 2971 | Oct | 1 | 17:10 | uz_parms.i |
| -r--r--r-- | 1 | root | other | 55015 | Oct | 1 | 17:10 | uzflow.f |
| -r--r--r-- | 1 | root | other | 100474 | Oct | 1 | 17:10 | uzft.f |
| -r--r--r-- | 1 | root | other | 10398 | Oct | 1 | 17:10 | volcano.f |

tpa_v3.1.1/SCCS:

total 2428

| | | | | | | | | |
|------------|---|------|-------|--------|-----|---|-------|--------------|
| -r--r--r-- | 1 | root | other | 1389 | Oct | 1 | 17:09 | s.CLEANUP |
| -r--r--r-- | 1 | root | other | 837 | Oct | 1 | 17:09 | s.Makefile |
| -r--r--r-- | 1 | root | other | 4338 | Oct | 1 | 17:09 | s.README |
| -r--r--r-- | 1 | root | other | 28902 | Oct | 1 | 17:09 | s.array.f |
| -r--r--r-- | 1 | root | other | 18054 | Oct | 1 | 17:09 | s.ashplumo.f |
| -r--r--r-- | 1 | root | other | 24001 | Oct | 1 | 17:09 | s.ashrmovo.f |
| -r--r--r-- | 1 | root | other | 9894 | Oct | 1 | 17:09 | s.condxzyt.f |
| -r--r--r-- | 1 | root | other | 15405 | Oct | 1 | 17:09 | s.dcags.f |
| -r--r--r-- | 1 | root | other | 21794 | Oct | 1 | 17:09 | s.dcagw.f |
| -r--r--r-- | 1 | root | other | 31323 | Oct | 1 | 17:09 | s.ebsfail.f |
| -r--r--r-- | 1 | root | other | 40527 | Oct | 1 | 17:09 | s.ebsrel.f |
| -r--r--r-- | 1 | root | other | 198398 | Oct | 1 | 17:09 | s.exec.f |
| -r--r--r-- | 1 | root | other | 1643 | Oct | 1 | 17:09 | s.execa.i |
| -r--r--r-- | 1 | root | other | 634 | Oct | 1 | 17:09 | s.execb.i |
| -r--r--r-- | 1 | root | other | 11055 | Oct | 1 | 17:09 | s.faulto.f |

9/45

| | | | | | | | | |
|------------|---|------|-------|-------|-----|---|-------|----------------|
| -r--r--r-- | 1 | root | othe_ | 6090 | Oct | 1 | 17:09 | s.fileu...t.f |
| -r--r--r-- | 1 | root | other | 5973 | Oct | 1 | 17:09 | s.findelev.f |
| -r--r--r-- | 1 | root | other | 208 | Oct | 1 | 17:09 | s.fu1.i |
| -r--r--r-- | 1 | root | other | 578 | Oct | 1 | 17:09 | s.fu2.i |
| -r--r--r-- | 1 | root | other | 55213 | Oct | 1 | 17:09 | s.invent.f |
| -r--r--r-- | 1 | root | other | 205 | Oct | 1 | 17:09 | s.inventa.i |
| -r--r--r-- | 1 | root | other | 257 | Oct | 1 | 17:09 | s.inventb.i |
| -r--r--r-- | 1 | root | other | 342 | Oct | 1 | 17:09 | s.inventc.i |
| -r--r--r-- | 1 | root | other | 210 | Oct | 1 | 17:09 | s.inventd.i |
| -r--r--r-- | 1 | root | other | 214 | Oct | 1 | 17:09 | s.invente.i |
| -r--r--r-- | 1 | root | other | 213 | Oct | 1 | 17:09 | s.inventf.i |
| -r--r--r-- | 1 | root | other | 212 | Oct | 1 | 17:09 | s.inventg.i |
| -r--r--r-- | 1 | root | other | 212 | Oct | 1 | 17:09 | s.inventh.i |
| -r--r--r-- | 1 | root | other | 223 | Oct | 1 | 17:09 | s.inventi.i |
| -r--r--r-- | 1 | root | other | 220 | Oct | 1 | 17:09 | s.inventj.i |
| -r--r--r-- | 1 | root | other | 254 | Oct | 1 | 17:09 | s.inventk.i |
| -r--r--r-- | 1 | root | other | 223 | Oct | 1 | 17:09 | s.inventl.i |
| -r--r--r-- | 1 | root | other | 226 | Oct | 1 | 17:09 | s.inventm.i |
| -r--r--r-- | 1 | root | other | 252 | Oct | 1 | 17:09 | s.inventn.i |
| -r--r--r-- | 1 | root | other | 226 | Oct | 1 | 17:09 | s.max500yr.i |
| -r--r--r-- | 1 | root | other | 247 | Oct | 1 | 17:09 | s.maxchain.i |
| -r--r--r-- | 1 | root | other | 334 | Oct | 1 | 17:09 | s.maxnnucl.i |
| -r--r--r-- | 1 | root | other | 378 | Oct | 1 | 17:09 | s.maxnsuba.i |
| -r--r--r-- | 1 | root | other | 278 | Oct | 1 | 17:09 | s.maxntime.i |
| -r--r--r-- | 1 | root | other | 11924 | Oct | 1 | 17:09 | s.mv.f |
| -r--r--r-- | 1 | root | other | 259 | Oct | 1 | 17:09 | s.mva.i |
| -r--r--r-- | 1 | root | other | 204 | Oct | 1 | 17:09 | s.mvb.i |
| -r--r--r-- | 1 | root | other | 205 | Oct | 1 | 17:09 | s.mvc.i |
| -r--r--r-- | 1 | root | other | 249 | Oct | 1 | 17:09 | s.mvd.i |
| -r--r--r-- | 1 | root | other | 220 | Oct | 1 | 17:09 | s.mve.i |
| -r--r--r-- | 1 | root | other | 220 | Oct | 1 | 17:09 | s.mvf.i |
| -r--r--r-- | 1 | root | other | 52519 | Oct | 1 | 17:09 | s.nfenv.f |
| -r--r--r-- | 1 | root | other | 242 | Oct | 1 | 17:09 | s.nintv.i |
| -r--r--r-- | 1 | root | other | 1650 | Oct | 1 | 17:09 | s.notice.i |
| -r--r--r-- | 1 | root | other | 4958 | Oct | 1 | 17:09 | s.numrecip.f |
| -r--r--r-- | 1 | root | other | 407 | Oct | 1 | 17:09 | s.path.i |
| -r--r--r-- | 1 | root | other | 6464 | Oct | 1 | 17:09 | s.peakfinder.f |
| -r--r--r-- | 1 | root | other | 44535 | Oct | 1 | 17:09 | s.ran.f |
| -r--r--r-- | 1 | root | other | 49014 | Oct | 1 | 17:09 | s.reader.f |
| -r--r--r-- | 1 | root | other | 206 | Oct | 1 | 17:09 | s.reflux2.i |
| -r--r--r-- | 1 | root | other | 84789 | Oct | 1 | 17:09 | s.sampler.f |
| -r--r--r-- | 1 | root | other | 210 | Oct | 1 | 17:09 | s.sampler0.i |
| -r--r--r-- | 1 | root | other | 227 | Oct | 1 | 17:09 | s.sampler1.i |
| -r--r--r-- | 1 | root | other | 210 | Oct | 1 | 17:09 | s.sampler2.i |
| -r--r--r-- | 1 | root | other | 210 | Oct | 1 | 17:09 | s.sampler.a.i |
| -r--r--r-- | 1 | root | other | 210 | Oct | 1 | 17:09 | s.samplerb.i |
| -r--r--r-- | 1 | root | other | 210 | Oct | 1 | 17:09 | s.samplerc.i |
| -r--r--r-- | 1 | root | other | 216 | Oct | 1 | 17:09 | s.samplerd.i |
| -r--r--r-- | 1 | root | other | 281 | Oct | 1 | 17:09 | s.sampler.e.i |
| -r--r--r-- | 1 | root | other | 259 | Oct | 1 | 17:09 | s.samplerf.i |
| -r--r--r-- | 1 | root | other | 232 | Oct | 1 | 17:09 | s.samplerg.i |
| -r--r--r-- | 1 | root | other | 216 | Oct | 1 | 17:09 | s.samplerh.i |
| -r--r--r-- | 1 | root | other | 231 | Oct | 1 | 17:09 | s.sampleri.i |
| -r--r--r-- | 1 | root | other | 209 | Oct | 1 | 17:09 | s.samplerj.i |
| -r--r--r-- | 1 | root | other | 356 | Oct | 1 | 17:09 | s.samplerk.i |
| -r--r--r-- | 1 | root | other | 252 | Oct | 1 | 17:09 | s.samplerl.i |
| -r--r--r-- | 1 | root | other | 211 | Oct | 1 | 17:09 | s.samplerm.i |
| -r--r--r-- | 1 | root | other | 227 | Oct | 1 | 17:09 | s.sampler.n.i |
| -r--r--r-- | 1 | root | other | 211 | Oct | 1 | 17:09 | s.sampler.o.i |
| -r--r--r-- | 1 | root | other | 268 | Oct | 1 | 17:09 | s.samplerp.i |
| -r--r--r-- | 1 | root | other | 251 | Oct | 1 | 17:09 | s.samplerq.i |
| -r--r--r-- | 1 | root | other | 324 | Oct | 1 | 17:10 | s.sampler.r.i |
| -r--r--r-- | 1 | root | other | 372 | Oct | 1 | 17:10 | s.sampler.s.i |
| -r--r--r-- | 1 | root | other | 218 | Oct | 1 | 17:10 | s.sampler.t.i |
| -r--r--r-- | 1 | root | other | 217 | Oct | 1 | 17:10 | s.sampler.u.i |
| -r--r--r-- | 1 | root | other | 210 | Oct | 1 | 17:10 | s.sampler.v.i |

10/45

| | | | | | | | | |
|------------|---|------|-------|--------|-----|---|-------|----------------------|
| -r--r--r-- | 1 | root | other | 208 | Oct | 1 | 17:10 | s.samplerw.i |
| -r--r--r-- | 1 | root | other | 238 | Oct | 1 | 17:10 | s.samplerx.i |
| -r--r--r-- | 1 | root | other | 208 | Oct | 1 | 17:10 | s.samplery.i |
| -r--r--r-- | 1 | root | other | 208 | Oct | 1 | 17:10 | s.samplerz.i |
| -r--r--r-- | 1 | root | other | 38652 | Oct | 1 | 17:10 | s.seismo.f |
| -r--r--r-- | 1 | root | other | 1856 | Oct | 1 | 17:10 | s.setfiles.i |
| -r--r--r-- | 1 | root | other | 292 | Oct | 1 | 17:10 | s.stop.i |
| -r--r--r-- | 1 | root | other | 35259 | Oct | 1 | 17:10 | s.subarea.f |
| -r--r--r-- | 1 | root | other | 282 | Oct | 1 | 17:10 | s.subareaa.i |
| -r--r--r-- | 1 | root | other | 227 | Oct | 1 | 17:10 | s.subareab.i |
| -r--r--r-- | 1 | root | other | 230 | Oct | 1 | 17:10 | s.subareac.i |
| -r--r--r-- | 1 | root | other | 229 | Oct | 1 | 17:10 | s.subaread.i |
| -r--r--r-- | 1 | root | other | 225 | Oct | 1 | 17:10 | s.subareae.i |
| -r--r--r-- | 1 | root | other | 57887 | Oct | 1 | 17:10 | s.szft.f |
| -r--r--r-- | 1 | root | other | 208 | Oct | 1 | 17:10 | s.szft.i |
| -r--r--r-- | 1 | root | other | 43526 | Oct | 1 | 17:10 | s.tpa.inp |
| -r--r--r-- | 1 | root | other | 45115 | Oct | 1 | 17:10 | s.tpa.inp.meanvalues |
| -r--r--r-- | 1 | root | other | 462 | Oct | 1 | 17:10 | s.uz_climi.i |
| -r--r--r-- | 1 | root | other | 1074 | Oct | 1 | 17:10 | s.uz_climr.i |
| -r--r--r-- | 1 | root | other | 489 | Oct | 1 | 17:10 | s.uz_climz.i |
| -r--r--r-- | 1 | root | other | 575 | Oct | 1 | 17:10 | s.uz_flowi.i |
| -r--r--r-- | 1 | root | other | 1029 | Oct | 1 | 17:10 | s.uz_flowr.i |
| -r--r--r-- | 1 | root | other | 324 | Oct | 1 | 17:10 | s.uz_flowz.i |
| -r--r--r-- | 1 | root | other | 3119 | Oct | 1 | 17:10 | s.uz_parms.i |
| -r--r--r-- | 1 | root | other | 55163 | Oct | 1 | 17:10 | s.uzflow.f |
| -r--r--r-- | 1 | root | other | 100622 | Oct | 1 | 17:10 | s.uzft.f |
| -r--r--r-- | 1 | root | other | 10546 | Oct | 1 | 17:10 | s.volcano.f |

tpa_v3.1.1/ccdf:

total 42

| | | | | | | | | |
|------------|---|------|-------|-------|-----|---|-------|-----------|
| -r--r--r-- | 1 | root | other | 174 | Oct | 1 | 17:08 | Makefile |
| drwxr-xr-x | 2 | root | other | 512 | Oct | 1 | 17:08 | SCCS |
| -r--r--r-- | 1 | root | other | 17138 | Oct | 1 | 17:08 | tccdf.f |
| -r--r--r-- | 1 | root | other | 66 | Oct | 1 | 17:08 | tccdf.i |
| -r--r--r-- | 1 | root | other | 640 | Oct | 1 | 17:09 | tccdf.inp |

tpa_v3.1.1/ccdf/SCCS:

total 40

| | | | | | | | | |
|------------|---|------|-------|-------|-----|---|-------|-------------|
| -r--r--r-- | 1 | root | other | 322 | Oct | 1 | 17:08 | s.Makefile |
| -r--r--r-- | 1 | root | other | 17286 | Oct | 1 | 17:08 | s.tccdf.f |
| -r--r--r-- | 1 | root | other | 214 | Oct | 1 | 17:08 | s.tccdf.i |
| -r--r--r-- | 1 | root | other | 788 | Oct | 1 | 17:08 | s.tccdf.inp |

tpa_v3.1.1/codes:

total 1638

| | | | | | | | | |
|------------|---|------|-------|--------|-----|---|-------|------------|
| -r--r--r-- | 1 | root | other | 306 | Oct | 1 | 17:07 | Makefile |
| -r--r--r-- | 1 | root | other | 499 | Oct | 1 | 17:07 | README |
| drwxr-xr-x | 2 | root | other | 512 | Oct | 1 | 17:07 | SCCS |
| -r--r--r-- | 1 | root | other | 2203 | Oct | 1 | 17:07 | SIZES.INC |
| -r--r--r-- | 1 | root | other | 164 | Oct | 1 | 17:07 | SIZES2.INC |
| -r--r--r-- | 1 | root | other | 94534 | Oct | 1 | 17:07 | ashplume.f |
| -r--r--r-- | 1 | root | other | 80381 | Oct | 1 | 17:07 | fault.f |
| -r--r--r-- | 1 | root | other | 301211 | Oct | 1 | 17:07 | nefmks.f |
| -r--r--r-- | 1 | root | other | 151436 | Oct | 1 | 17:07 | releaset.f |
| -r--r--r-- | 1 | root | other | 171358 | Oct | 1 | 17:07 | snllhs.f |

tpa_v3.1.1/codes/SCCS:

total 1636

| | | | | | | | | |
|------------|---|------|-------|--------|-----|---|-------|--------------|
| -r--r--r-- | 1 | root | other | 454 | Oct | 1 | 17:07 | s.Makefile |
| -r--r--r-- | 1 | root | other | 647 | Oct | 1 | 17:07 | s.README |
| -r--r--r-- | 1 | root | other | 2351 | Oct | 1 | 17:07 | s.SIZES.INC |
| -r--r--r-- | 1 | root | other | 312 | Oct | 1 | 17:07 | s.SIZES2.INC |
| -r--r--r-- | 1 | root | other | 94682 | Oct | 1 | 17:07 | s.ashplume.f |
| -r--r--r-- | 1 | root | other | 80529 | Oct | 1 | 17:07 | s.fault.f |
| -r--r--r-- | 1 | root | other | 301359 | Oct | 1 | 17:07 | s.nefmks.f |
| -r--r--r-- | 1 | root | other | 151584 | Oct | 1 | 17:07 | s.releaset.f |

11/45

```
-r--r--r-- 1 root other 171506 Oct 1 17:07 s.snllh.f
```

tpa_v3.1.1/data:

total 7600

```
drwxr-xr-x 2 root other 1024 Oct 1 17:08 SCCS
-r--r--r-- 1 root other 850000 Oct 1 17:08 climato1.dat
-r--r--r-- 1 root other 2460 Oct 1 17:08 climato2.dat
-r--r--r-- 1 root other 4586 Oct 1 17:08 ebsfail.def
-r--r--r-- 1 root other 1648 Oct 1 17:08 ebspac.nuc
-r--r--r-- 1 root other 3176 Oct 1 17:08 ebsrel.def
-r--r--r-- 1 root other 301091 Oct 1 17:08 elevdem.dat
-r--r--r-- 1 root other 1884 Oct 1 17:08 gs_cb_ad.dat
-r--r--r-- 1 root other 1061 Oct 1 17:08 gs_cb_ci.dat
-r--r--r-- 1 root other 2052 Oct 1 17:08 gs_pb_ad.dat
-r--r--r-- 1 root other 1145 Oct 1 17:08 gs_pb_ci.dat
-r--r--r-- 1 root other 2483 Oct 1 17:08 gw_cb_ad.dat
-r--r--r-- 1 root other 610 Oct 1 17:08 gw_cb_ci.dat
-r--r--r-- 1 root other 2718 Oct 1 17:08 gw_pb_ad.dat
-r--r--r-- 1 root other 651 Oct 1 17:08 gw_pb_ci.dat
-r--r--r-- 1 root other 1274984 Oct 1 17:08 multiflo.dat
-r--r--r-- 1 root other 2260 Oct 1 17:08 rectedge.dat
-r--r--r-- 1 root other 544 Oct 1 17:08 seismo.dat
-r--r--r-- 1 root other 494027 Oct 1 17:08 soildem.dat
-r--r--r-- 1 root other 3833 Oct 1 17:08 strmtube.dat
-r--r--r-- 1 root other 754061 Oct 1 17:08 tefkti.inp
-r--r--r-- 1 root other 103334 Oct 1 17:08 tpanames.dbs
```

tpa_v3.1.1/data/SCCS:

total 7598

```
-r--r--r-- 1 root other 850148 Oct 1 17:08 s.climato1.dat
-r--r--r-- 1 root other 2608 Oct 1 17:08 s.climato2.dat
-r--r--r-- 1 root other 4734 Oct 1 17:08 s.ebsfail.def
-r--r--r-- 1 root other 1796 Oct 1 17:08 s.ebspac.nuc
-r--r--r-- 1 root other 3324 Oct 1 17:08 s.ebsrel.def
-r--r--r-- 1 root other 301239 Oct 1 17:08 s.elevdem.dat
-r--r--r-- 1 root other 2032 Oct 1 17:08 s.gs_cb_ad.dat
-r--r--r-- 1 root other 1209 Oct 1 17:08 s.gs_cb_ci.dat
-r--r--r-- 1 root other 2200 Oct 1 17:08 s.gs_pb_ad.dat
-r--r--r-- 1 root other 1293 Oct 1 17:08 s.gs_pb_ci.dat
-r--r--r-- 1 root other 2631 Oct 1 17:08 s.gw_cb_ad.dat
-r--r--r-- 1 root other 758 Oct 1 17:08 s.gw_cb_ci.dat
-r--r--r-- 1 root other 2866 Oct 1 17:08 s.gw_pb_ad.dat
-r--r--r-- 1 root other 799 Oct 1 17:08 s.gw_pb_ci.dat
-r--r--r-- 1 root other 1275132 Oct 1 17:08 s.multiflo.dat
-r--r--r-- 1 root other 2408 Oct 1 17:08 s.rectedge.dat
-r--r--r-- 1 root other 692 Oct 1 17:08 s.seismo.dat
-r--r--r-- 1 root other 494175 Oct 1 17:08 s.soildem.dat
-r--r--r-- 1 root other 3981 Oct 1 17:08 s.strmtube.dat
-r--r--r-- 1 root other 754209 Oct 1 17:08 s.tefkti.inp
-r--r--r-- 1 root other 103482 Oct 1 17:08 s.tpanames.dbs
```

12/45

SOFTWARE PROBLEM/CHANGE REPORT

| | | |
|---|----------------------------------|--------------------------------|
| 01. SPCR Number: PA-SPCR-204 | | |
| 02. Project Title: TPA Code Version 3.1 | | 20-5708-762 Project Number: |
| 03. SPCR Title: Incorrect Index for array RHO in subroutine TRNSPT | | |
| 04. AFFECTED SOFTWARE AND/OR DOCUMENTATION | | |
| nefmks.f | | |
| 05. DESCRIPTION OF PROBLEM/CHANGE | | |
| Array out of bound error | | |
| 06. PROBLEM SOLUTION/NEED FOR CHANGE | | |
| Please see attached | | |
| 07. Originator: T. McCashin | Title: NRC | Date: 9/19/97 |
| PROJECT | | |
| 08. Need by Date: | 09. Approved: [Signature] | Disapproved: (10/1/97) |
| 10. Software Developer: S. Mohanty et al. | Date: 9/19/97 | |
| 11. Element Manager: S. Mohanty for R. Baca | Date: 10/1/97 | |
| 12. IMPLEMENTED SOLUTION | | |
| Please see attached | | |
| 13. Implemented By: Original implementation by R. Codell | | Date: 9/19/97 |

postscript

JOB 903

SPCR.003

For: Sitakanta Mohanty
Creator: Microsoft Word: LaserWriter 8 8.3.4
Creation Date: 5:32 PM Friday, September 19, 1997

Submit queue: IF 1 / Ethernet / ETHERTALK
Submitted: Thu Sep 19 17:38:11 1997
Started: Thu Sep 19 17:38:12 1997



QMS 3825 Print System

QMS 3825 Print System

01. SPCR Number: 3

02. Project Title: TPA Code
Project Number: NA

03. SPCR Title: Incorrect index for array RHO in
subroutine TRNSPT

04. AFFECTED SOFTWARE AND/OR DOCUMENTATION

NEFTRAN II - software only

05. DESCRIPTION OF PROBLEM/CHANGE

The subroutine TRNSPT loops over the RHO array (array that contains the density of atoms) in a calculation that determines the amount of mass that discharges in a given time step. For certain chains and retardation values the index on RHO attains a negative or zero value (the occurrence is tied to having the last member of a chain with a short half-life such that it is not considered important in NEFTRAN II and thus does not affect the selection of a time step when this "unimportant" nuclide is the least retarded nuclide in the chain it goes further than the other members in the chain but is not used in determining the time step). The error arises when the increment for the RHO array is determined in subroutine SETDIS by evaluating the furthest distance that the last member in the chain can travel. If the largest velocity value in the velocity distribution results in a travel time to discharge which is less than the time step then the calculation results in a path length that is larger than that specified in the problem input - this results in undefined array locations being addressed in the RHO array.

06. PROBLEM SOLUTION/NEED FOR CHANGE

The solution to this problem is to perform a check on the calculation and not allow the travel distance to exceed the problem size which in the code is defined as NTX (total number of grid blocks).

In Subroutine SETDIS

old line: $L = V(NVI, KP, NPT) * DT / DX(NPT) + 1$

correction: $L = V(NVI, KP, NPT) * DT / DX(NPT) + 1$
IF (L .GT. NTX) L=NTX

07. Originator:

Tim McCartin

Title:

Analyst

Date: 9/17/97

PROJECT

08. Need by Date:

09. Approved:

Disapproved:

10. Software Developer:

Date:

11. Element Manager:

Date:

12. IMPLEMENTED SOLUTION

13. Implemented By:

Date:

4172a4173,4176

```
> c rwj 9/18/97 Add line to prevent 0 subscript in RHO array for last member
> c in a chain of 3 or more.
> IF (L .GT. NTX) L = NTX
> c
```

d:ff nefmks.f nefmksrho.f

listing of diff script.

```
echo diff tpa.inp
diff tpa.inp $1
echo diff tpa.inp.meanvalues
diff tpa.inp.meanvalues $1
echo diff NEFII.VEL
diff NEFII.VEL $1
echo diff ashplume.in
diff ashplume.in $1
echo diff ashplume.out
diff ashplume.out $1
echo diff ashplumo.ech
diff ashplumo.ech $1
echo diff ashplumo.res
diff ashplumo.res $1
echo diff ashrmovo.ech
diff ashrmovo.ech $1
echo diff ashrmovo.res
diff ashrmovo.res $1
echo diff chloridemf.dat
diff chloridemf.dat $1
echo diff climato1.dat
diff climato1.dat $1
echo diff climato2.dat
diff climato2.dat $1
echo diff climato.inp
diff climato.inp $1
echo diff corrode.out
diff corrode.out $1
echo diff cp.tpa
diff cp.tpa $1
echo diff cumcl4.out
diff cumcl4.out $1
echo diff dcags.ech
diff dcags.ech $1
echo diff dcags.res
diff dcags.res $1
echo diff dcagw.ech
diff dcagw.ech $1
echo diff dcagw.res
diff dcagw.res $1
echo diff dagnostic.out
diff dagnostic.out $1
echo diff ebsfail.ech
diff ebsfail.ech $1
echo diff ebsfail.inp
diff ebsfail.inp $1
echo diff ebsfail.res
diff ebsfail.res $1
echo diff ebsfail.tmp
diff ebsfail.tmp $1
echo diff ebsflo.dat
diff ebsflo.dat $1
# echo diff ebsnef.dat
# diff ebsnef.dat $1
echo diff ebsnuc.dat
diff ebsnuc.dat $1
echo diff ebspac.nuc
diff ebspac.nuc $1
echo diff ebsrel.ech
diff ebsrel.ech $1
echo diff ebsrel.inp
diff ebsrel.inp $1
echo diff ebsrel.res
diff ebsrel.res $1
echo diff ebsrel.tmp
```

```
diff ebsrel.tmp $1
echo diff ebstrh.dat
diff ebstrh.dat $1
echo diff ebstrhc.inp
diff ebstrhc.inp $1
echo diff echo_fail.dat
diff echo_fail.dat $1
echo diff echo_release.out
diff echo_release.out $1
echo diff gwccdf.res
diff gwccdf.res $1
echo diff gsccdf.res
diff gsccdf.res $1
echo diff relccdf.res
diff relccdf.res $1
echo diff relgwgs.res
diff relgwgs.res $1
echo diff totdose.res
diff totdose.res $1
echo diff fault.out
diff fault.out $1
echo diff faulto.ech
diff faulto.ech $1
echo diff faulto.res
diff faulto.res $1
echo diff infile.ash
diff infile.ash $1
echo diff invl000.out
diff invl000.out $1
echo diff junk.out
diff junk.out $1
echo diff maxrel.dat
diff maxrel.dat $1
echo diff multiflo.dat
diff multiflo.dat $1
echo diff nefii.inp
diff nefii.inp $1
echo diff nefiisz.inp
diff nefiisz.inp $1
echo diff nefiiuz.inp
diff nefiiuz.inp $1
echo diff nefiiuz.vel
diff nefiiuz.vel $1
echo diff nfenv.ech
diff nfenv.ech $1
echo diff nfenv.res
diff nfenv.res $1
echo diff ratecl4.out
diff ratecl4.out $1
echo diff rectedge.dat
diff rectedge.dat $1
echo diff relcum.out
diff relcum.out $1
echo diff release.out
diff release.out $1
echo diff releaset.out
diff releaset.out $1
echo diff relfrac.out
diff relfrac.out $1
echo diff rgsna.tpa
diff rgsna.tpa $1
echo diff rgsnr.tpa
diff rgsnr.tpa $1
echo diff rgssa.tpa
diff rgssa.tpa $1
```

19/45

echo diff rgssr.tpa
diff rgssr.tpa \$1
echo diff rgwgssa.tpa
diff rgwgssa.tpa \$1
echo diff rgwna.tpa
diff rgwna.tpa \$1
echo diff rgwnr.tpa
diff rgwnr.tpa \$1
echo diff rgwsa.tpa
diff rgwsa.tpa \$1
echo diff rgwsr.tpa
diff rgwsr.tpa \$1
echo diff seismo.ech
diff seismo.ech \$1
echo diff seismo.res
diff seismo.res \$1
echo diff soildep.dem
diff soildep.dem \$1
echo diff sotnef.dat
diff sotnef.dat \$1
echo diff sp.tpa
diff sp.tpa \$1
echo diff spquery.tpa
diff spquery.tpa \$1
echo diff szft.ech
diff szft.ech \$1
echo diff szft.res
diff szft.res \$1
echo diff temphumd.dat
diff temphumd.dat \$1
echo diff transition.out
diff transition.out \$1
echo diff treleasel.out
diff treleasel.out \$1
echo diff uzflow.ech
diff uzflow.ech \$1
echo diff uzflow.res
diff uzflow.res \$1
echo diff uzft.ech
diff uzft.ech \$1
echo diff uzft.res
diff uzft.res \$1
echo diff volcano.ech
diff volcano.ech \$1
echo diff volcano.res
diff volcano.res \$1
echo diff watrel.in
diff watrel.in \$1
echo diff ymelev.dem
diff ymelev.dem \$1
echo diff nefii.out
diff nefii.out \$1
echo diff nefii.rel
diff nefii.rel \$1
echo diff nefiisz.out
diff nefiisz.out \$1
echo diff nefiisz.src
diff nefiisz.src \$1
echo diff nefiiuz.out
diff nefiiuz.out \$1
echo diff nefiiuz.src
diff nefiiuz.src \$1
echo diff mv.tpa
diff mv.tpa \$1
echo airpkdos.res

```
diff airpkdos.res $1
echo ashout.res
diff ashout.res $1
echo ccdfgwgs.res
diff ccdfgwgs.res $1
echo cumrel.res
diff cumrel.res $1
echo cumrelease.out
diff cumrelease.out $1
echo gwpkdos.res
diff gwpkdos.res $1
echo gwttuusz.res
diff gwttuusz.res $1
echo infilper.res
diff infilper.res $1
echo nearfld.res
diff nearfld.res $1
echo npkdoset.res
diff npkdoset.res $1
echo pkreltim.res
diff pkreltim.res $1
echo samplpar.res
diff samplpar.res $1
echo totdose.res
diff totdose.res $1
echo wpsfail.res
diff wpsfail.res $1
echo lhs.inp
diff lhs.inp $1
echo lhs.out
diff lhs.out $1
echo samplpar.hdr
diff samplpar.hdr $1
echo samplpar.abb
diff samplpar.abb $1
echo tpa.out
diff tpa.out $1
```

output of diff script.

```
diff tpa.inp
diff tpa.inp.meanvalues
diff NEFII.VEL
diff ashplume.in
diff ashplume.out
diff ashplumo.ech
diff ashplumo.res
diff ashrmovo.ech
diff ashrmovo.res
diff chloridemf.dat
diff climatol.dat
diff climato2.dat
diff climato.inp
diff corrode.out
diff cp.tpa
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
diff cumcl4.out
diff dcags.ech
diff dcags.res
diff dcagw.ech
diff dcagw.res
diff dignostic.out
diff ebsfail.ech
diff ebsfail.inp
diff ebsfail.res
diff ebsfail.tmp
diff ebsflo.dat
diff ebsnuc.dat
diff ebspac.nuc
diff ebsrel.ech
diff ebsrel.inp
diff ebsrel.res
diff ebsrel.tmp
diff ebstrh.dat
diff ebstrhc.inp
diff echo_fail.dat
diff echo_release.out
diff gwccdf.res
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
diff gsccdf.res
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
diff relccdf.res
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
diff relgwgs.res
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
diff totdose.res
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
```

```
diff failt.out
4c4
< Fri Sep 19 09:20:35 1997
---
> Thu Sep 18 14:34:18 1997
diff faulto.ech
diff faulto.res
diff infile.ash
diff invl000.out
diff junk.out
diff maxrel.dat
diff multiflo.dat
diff nefii.inp
diff nefiisz.inp
diff nefiiuz.inp
diff nefiiuz.vel
diff nfenv.ech
diff nfenv.res
diff ratecl4.out
diff rectedge.dat
diff relcum.out
diff release.out
diff releaset.out
3c3
< Fri Sep 19 09:20:38 1997
---
> Thu Sep 18 14:34:21 1997
diff relfrac.out
diff rgsna.tpa
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
diff rgsnr.tpa
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
diff rgssa.tpa
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
diff rgssr.tpa
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
diff rgwgssa.tpa
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
diff rgwna.tpa
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
diff rgwnr.tpa
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
diff rgwsa.tpa
3c3
```

```

< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
diff rgwsr.tpa
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
diff seismo.ech
diff seismo.res
diff soildep.dem
diff sotnef.dat
diff sp.tpa
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
diff spquery.tpa
diff szft.ech
diff szft.res
diff temphumd.dat
diff transition.out
diff treleasel.out
diff uzflow.ech
diff uzflow.res
diff uzft.ech
diff uzft.res
diff volcano.ech
diff volcano.res
diff watrel.in
diff ymelev.dem
diff nefii.out
9c9
<      *      EXECUTION DATE Sep 19   AND TIME 09:20:4      *
---
>      *      EXECUTION DATE Sep 18   AND TIME 14:34:3      *
diff nefii.rel
diff nefiisz.out
9c9
<      *      EXECUTION DATE Sep 19   AND TIME 09:20:4      *
---
>      *      EXECUTION DATE Sep 18   AND TIME 14:34:3      *
diff nefiisz.src
diff nefiiuz.out
9c9
<      *      EXECUTION DATE Sep 19   AND TIME 09:20:1      *
---
>      *      EXECUTION DATE Sep 18   AND TIME 14:34:0      *
diff nefiiuz.src
diff mv.tpa
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
airpkdos.res
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
ashout.res
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
ccdfgws.res

```

```
cumrel.res
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
cumrelease.out
gwpkdos.res
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
gwttuusz.res
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
infilper.res
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
nearfld.res
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
npkdoset.res
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
pkreltim.res
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
samplpar.res
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
totdose.res
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
wpsfail.res
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
lhs.inp
lhs.out
samplpar.hdr
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
samplpar.abb
3c3
< TPA 3.1, Job started: Fri Sep 19 09:05:20 1997
---
> TPA 3.1, Job started: Thu Sep 18 14:19:51 1997
tpa.out
```


listing of
files in
ssv3r7a

```

total 11204
drwxr-xr-x  2 janetzke sunuser    2048 Sep 18 14:44 .
drwx----- 22 janetzke sunuser   7168 Sep 19 17:22 ..
-rw-r--r--  1 janetzke sunuser     93 Sep 18 14:34 NEFII.VEL
-rw-r--r--  1 janetzke sunuser   4209 Sep 18 14:34 airpkdos.res
-rw-r--r--  1 janetzke sunuser   1161 Sep 18 14:34 ashout.res
-rwxr-xr-x  1 janetzke sunuser  125668 Sep 18 14:27 ashplume.e
-rw-r--r--  1 janetzke sunuser   1550 Sep 18 14:34 ashplume.in
-rw-r--r--  1 janetzke sunuser   2041 Sep 18 14:34 ashplume.out
-rw-r--r--  1 janetzke sunuser   8674 Sep 18 14:34 chloridemf.dat
-rw-r--r--  1 janetzke sunuser  850000 Sep 18 14:20 climatol.dat
-rw-r--r--  1 janetzke sunuser   2460 Sep 18 14:20 climato2.dat
-rw-r--r--  1 janetzke sunuser   14506 Sep 18 14:34 corrode.out
-rw-r--r--  1 janetzke sunuser   48503 Sep 18 14:34 cp.tpa
-rw-r--r--  1 janetzke sunuser   6927 Sep 18 14:34 cumcl4.out
-rw-r--r--  1 janetzke sunuser   5367 Sep 18 14:34 cumrel.res
-rw-r--r--  1 janetzke sunuser   46580 Sep 18 14:34 cumrelease.out
-rw-r--r--  1 janetzke sunuser   9800 Sep 18 14:34 dignostic.out
-rw-r--r--  1 janetzke sunuser   4697 Sep 18 14:34 ebsfail.inp
-rw-r--r--  1 janetzke sunuser   6392 Sep 18 14:34 ebsflo.dat
-rw-r--r--  1 janetzke sunuser  108203 Sep 18 14:34 ebsnef.dat
-rw-r--r--  1 janetzke sunuser   1648 Sep 18 14:20 ebsnuc.dat
-rw-r--r--  1 janetzke sunuser   4698 Sep 18 14:34 ebspac.nuc
-rw-r--r--  1 janetzke sunuser   6075 Sep 18 14:34 ebsrel.inp
-rw-r--r--  1 janetzke sunuser   10968 Sep 18 14:34 ebstrh.dat
-rw-r--r--  1 janetzke sunuser   9923 Sep 18 14:34 ebstrhc.inp
-rw-r--r--  1 janetzke sunuser   1943 Sep 18 14:34 echo_fail.dat
-rw-r--r--  1 janetzke sunuser  141723 Sep 18 14:34 echo_release.out
-rwxr-xr-x  1 janetzke sunuser   104084 Sep 18 14:20 fault.e
-rw-r--r--  1 janetzke sunuser   17302 Sep 18 14:34 fault.out
-rw-r--r--  1 janetzke sunuser   46580 Sep 18 14:34 frac_relrate.out
-rw-r--r--  1 janetzke sunuser    637 Sep 18 14:34 gsccdf.res
-rw-r--r--  1 janetzke sunuser    637 Sep 18 14:34 gwccdf.res
-rw-r--r--  1 janetzke sunuser   2529 Sep 18 14:34 gwpkdos.res
-rw-r--r--  1 janetzke sunuser   2529 Sep 18 14:34 gwtuzsz.res
-rw-r--r--  1 janetzke sunuser    540 Sep 18 14:34 infile.ash
-rw-r--r--  1 janetzke sunuser   5601 Sep 18 14:34 infilper.res
-rw-r--r--  1 janetzke sunuser   1103 Sep 18 14:34 invl000.out
-rw-r--r--  1 janetzke sunuser   3655 Sep 18 14:34 junk.out
-rw-r--r--  1 janetzke sunuser   25192 Sep 18 14:20 lhs.inp
-rw-r--r--  1 janetzke sunuser   10542 Sep 18 14:20 lhs.out
-rw-r--r--  1 janetzke sunuser   1218 Sep 18 14:34 maxrel.dat
-rw-r--r--  1 janetzke sunuser  1274984 Sep 18 14:20 multiflo.dat
-rw-r--r--  1 janetzke sunuser   61558 Sep 18 14:34 mv.tpa
-rw-r--r--  1 janetzke sunuser   5601 Sep 18 14:34 nearfld.res
-rw-r--r--  1 janetzke sunuser   14806 Sep 18 14:34 nefii.inp
-rw-r--r--  1 janetzke sunuser   49356 Sep 18 14:34 nefii.out
-rw-r--r--  1 janetzke sunuser    456 Sep 18 14:34 nefii.rel
-rw-r--r--  1 janetzke sunuser   14806 Sep 18 14:34 nefiisz.inp
-rw-r--r--  1 janetzke sunuser   49356 Sep 18 14:34 nefiisz.out
-rw-r--r--  1 janetzke sunuser  113379 Sep 18 14:34 nefiisz.src
-rw-r--r--  1 janetzke sunuser   8962 Sep 18 14:34 nefiiuz.inp
-rw-r--r--  1 janetzke sunuser   38414 Sep 18 14:34 nefiiuz.out
-rw-r--r--  1 janetzke sunuser  116799 Sep 18 14:34 nefiiuz.src
-rw-r--r--  1 janetzke sunuser     93 Sep 18 14:34 nefiiuz.vel
-rwxr-xr-x  1 janetzke sunuser   691232 Sep 18 14:20 nefmks.e
-rw-r--r--  1 janetzke sunuser   3809 Sep 18 14:34 npkdoset.res
-rw-r--r--  1 janetzke sunuser   16287 Sep 18 14:34 pkreltim.res
-rw-r--r--  1 janetzke sunuser   7413 Sep 18 14:34 ratecl4.out
-rw-r--r--  1 janetzke sunuser   2260 Sep 18 14:20 rectedge.dat
-rw-r--r--  1 janetzke sunuser    637 Sep 18 14:34 relccdf.res
-rw-r--r--  1 janetzke sunuser    718 Sep 18 14:34 relcum.out
-rwxr-xr-x  1 janetzke sunuser  115576 Sep 18 14:20 releaset.e
-rw-r--r--  1 janetzke sunuser   16340 Sep 18 14:34 releaset.out
-rw-r--r--  1 janetzke sunuser    620 Sep 18 14:34 relfrac.out

```

```
-rw-r--r-- 1 janetzke sunuser      841 Sep 18 14:34 relgws.res
-rw-r--r-- 1 janetzke sunuser    98662 Sep 18 14:34 rgsna.tpa
-rw-r--r-- 1 janetzke sunuser   294602 Sep 18 14:34 rgsnr.tpa
-rw-r--r-- 1 janetzke sunuser     5132 Sep 18 14:34 rgssa.tpa
-rw-r--r-- 1 janetzke sunuser    14470 Sep 18 14:34 rgssr.tpa
-rw-r--r-- 1 janetzke sunuser     5132 Sep 18 14:34 rgwgssa.tpa
-rw-r--r-- 1 janetzke sunuser    47396 Sep 18 14:34 rgwna.tpa
-rw-r--r-- 1 janetzke sunuser   141124 Sep 18 14:34 rgwnr.tpa
-rw-r--r-- 1 janetzke sunuser     5132 Sep 18 14:34 rgwsa.tpa
-rw-r--r-- 1 janetzke sunuser    14470 Sep 18 14:34 rgwsr.tpa
-rw-r--r-- 1 janetzke sunuser     2566 Sep 18 14:20 samplpar.abb
-rw-r--r-- 1 janetzke sunuser    18386 Sep 18 14:20 samplpar.hdr
-rw-r--r-- 1 janetzke sunuser     11183 Sep 18 14:34 samplpar.res
-rw-r--r-- 1 janetzke sunuser   113379 Sep 18 14:34 sotnef.dat
-rw-r--r-- 1 janetzke sunuser    28640 Sep 18 14:34 sp.tpa
-rw-r--r-- 1 janetzke sunuser     3833 Sep 18 14:21 strmtube.dat
-rw-r--r-- 1 janetzke sunuser    38041 Sep 18 14:34 totdose.res
-rw-r--r-- 1 janetzke sunuser    43382 Sep 18 14:44 tpa.inp
-rw-r--r-- 1 janetzke sunuser    44896 Sep 18 14:44 tpa.inp.basecase
-rw-r--r-- 1 janetzke sunuser    41519 Sep 18 14:34 tpa.out
-rw-r--r-- 1 janetzke sunuser   103334 Sep 18 14:19 tpanames.dbs
-rw-r--r-- 1 janetzke sunuser   146820 Sep 18 14:34 treleasel.out
-rw-r--r-- 1 janetzke sunuser    21389 Sep 18 14:34 watrel.in
-rw-r--r-- 1 janetzke sunuser     1753 Sep 18 14:34 wpsfail.res
```

27/45

SOFTWARE PROBLEM/CHANGE REPORT

| | | |
|--|----------------------------------|--------------------------------|
| 01. SPCR Number: AA-SPCR-205 | | |
| 02. Project Title: TPA3.1 | | 20-5708-762 Project Number: |
| 03. SPCR Title: Reduced Execution Memory Size | | |
| 04. AFFECTED SOFTWARE AND/OR DOCUMENTATION | | |
| nefmks.f, maxnsuba.i (used in ebstail.f, obsrel.f, exec.f, reader.f, subrsort.f, uzft.f), Makefile for nefmks.f. | | |
| 05. DESCRIPTION OF PROBLEM/CHANGE | | |
| nefmks.e is 59,091,622 bytes + exec.e is about 50 Mbytes. Together they require over 100 Mbytes of memory for each user running the code. When multiple users are running on the same machine insufficient memory may prevent users from executing the code. | | |
| 06. PROBLEM SOLUTION/NEED FOR CHANGE | | |
| <u>nefmks.f</u> Change all REAL*8 variables TO REAL and compile without the -r8 option. This will give a size of 29,523,772. <u>maxnsuba.i</u> Change parameter maxnsubarea TO 30 instead of 100. This will give a size of 17,925,146. | | |
| 07. Originator: R. Codell | Title: NRC | Date: 9-23-97 |
| PROJECT | | |
| 08. Need by Date: | 09. Approved: [Signature] | Disapproved: (10/1/97) |
| 10. Software Developer: Janetzke | Date: 9-30-97 | |
| 11. Element Manager: S. Mohanty for R. Bacon | Date: 10/1/97 | |
| 12. IMPLEMENTED SOLUTION | | |
| nefmks.f was replaced with nefmksf.f received via e-mail from S. Mohanty. maxnsuba.i was edited as shown above in section 6. Some nefmks results are different in the last digit, but are | | |
| 13. Implemented By: Ron Janetzke | Date: 9-30-97 | |

 Satisfied
 TO NRC.
 CR. Codell

cc:Mail for: Sitakanta Mohanty

Subject: proposed revisions to tpa 3.1
From: UD1.UP1:RBC at PSEUDO 1997/09/24 3:56 PM
To: Sitakanta Mohanty (SMOHANTY) at CNWRA
cc: TWD2.TWP7:kim at PSEUDO
cc: TWD2.TWP7:tjm3 at PSEUDO

Date: 09/24/1997 03:56 pm (Wednesday)
From: Richard Codell
To: SWRI.CNWRA-OS2.SMOHANTY
CC: tjm3,kim
Subject: proposed revisions to tpa 3.1

As we discussed, here are two proposed revisions to tpa 3.1 to improve the efficiency by reducing the memory requirements substantially. The first is a single-precision version of nefmks.f, which I call nefmkss.f. It must also be compiled without the -r8 switch in the makefile. The second change is change the include file maxnsuba.i to reduce the parameter maxnsubarea from 100 to something more reasonable like 7. These two changes should reduce the memory requirements by approximately half, greatly facilitating the ability to run on our Sun workstations. There are a number of other memory savings that could be justified, such as reducing the use of double precision arrays except where we need the accuracy, and reducing the size of the arrays to something more like the maximum we expect to run. I believe we can trim another 50% with little trouble.

Dick

29/45

```
138c138
< REAL*8 LAMBDA, C, AW, P
---
> REAL LAMBDA, C, AW, P
142c142
< REAL*8 PEAK(MXISO), TIMBRK(MXISO), TIMPK(MXISO), TOTD(MXISO),
---
> REAL PEAK(MXISO), TIMBRK(MXISO), TIMPK(MXISO), TOTD(MXISO),
147c147
< REAL*8 BCJ(MXJCT)
---
> REAL BCJ(MXJCT)
152c152
< REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
> REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
162c162
< REAL*8 PORE(MXPTH), DT, Y
---
> REAL PORE(MXPTH), DT, Y
169c169
< REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
> REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
337c337
< REAL*8 LAMBDA, C, AW, P
---
> REAL LAMBDA, C, AW, P
349c349
< REAL*8 PEAK(MXISO), TOTD(MXISO), WDIS(MXISO), WDISUM,
---
> REAL PEAK(MXISO), TOTD(MXISO), WDIS(MXISO), WDISUM,
354c354
< REAL*8 BCJ(MXJCT)
---
> REAL BCJ(MXJCT)
362c362
< REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
> REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
372c372
< REAL*8 TH(MXLEG), Q(MXLEG), PATH(MXLEG)
---
> REAL TH(MXLEG), Q(MXLEG), PATH(MXLEG)
376c376
< REAL*8 PORE(MXPTH), DT, Y
---
> REAL PORE(MXPTH), DT, Y
383c383
< REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
> REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
387c387
< REAL*8 TSRCE(MXQSC), QSRCE(MXQSC)
---
> REAL TSRCE(MXQSC), QSRCE(MXQSC)
393c393
< REAL*8 QSC, VOL
---
> REAL QSC, VOL
397c397
< REAL*8 TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
---
> REAL TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
401c401
```

```

<      REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
---
>      REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
406c406
<      REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
>      REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
1049,1050c1049,1050
<      REAL*8 MKSDX, MKSDXX, LAMBDA, TDNM
<      REAL*8 C0, C1, C2, D1, D2, D3, V1, V2, V3, DRPJ, CN, FMRPJ,
---
>      REAL MKSDX, MKSDXX, LAMBDA, TDNM
>      REAL C0, C1, C2, D1, D2, D3, V1, V2, V3, DRPJ, CN, FMRPJ,
1054c1054
<      REAL*8 TH(MXLEG), Q(MXLEG), PATH(MXLEG)
---
>      REAL TH(MXLEG), Q(MXLEG), PATH(MXLEG)
1061c1061
<      REAL*8 Curout(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
>      REAL Curout(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
1069c1069
<      REAL*8 PORE(MXPTH), DT, Y
---
>      REAL PORE(MXPTH), DT, Y
1076c1076
<      REAL*8 VOL, QSC
---
>      REAL VOL, QSC
1080c1080
<      REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
---
>      REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
1085c1085
<      REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
>      REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
1092c1092
<      REAL*8 GA(MXNVI), R, SIGIR, TNUM, VL(MXMEM,MXNVI)
---
>      REAL GA(MXNVI), R, SIGIR, TNUM, VL(MXMEM,MXNVI)
1458,1459c1458,1459
<      REAL*8 LAMBDA
<      REAL*8 DL, DM, TSI, DFR, DLIN, DR, DTIN, DSTSAV, SUM,
---
>      REAL LAMBDA
>      REAL DL, DM, TSI, DFR, DLIN, DR, DTIN, DSTSAV, SUM,
1462c1462
<      REAL*8 ABFX
---
>      REAL ABFX
1465c1465
<      REAL*8 Curout(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
>      REAL Curout(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
1473c1473
<      REAL*8 TH(MXLEG), PATH(MXLEG), Q(MXLEG)
---
>      REAL TH(MXLEG), PATH(MXLEG), Q(MXLEG)
1477c1477
<      REAL*8 PORE(MXPTH), DT, Y
---
>      REAL PORE(MXPTH), DT, Y
1484c1484
<      REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)

```

```

---
> REAL DX(MXPETH), TMN(MXTSP), BF(NDBF)
1489c1489
< REAL*8 V(MXNVI,MXSUB,MXPETH), DXX, B(2*MXNVI, MXSUB, MXPETH),
---
> REAL V(MXNVI,MXSUB,MXPETH), DXX, B(2*MXNVI, MXSUB, MXPETH),
1496c1496
< REAL*8 TS(MXMEM), VL(MXMEM,MXNVI), VA(MXMEM,MXNVI),
---
> REAL TS(MXMEM), VL(MXMEM,MXNVI), VA(MXMEM,MXNVI),
1920,1921c1920,1921
< REAL*8 DIS, DSMX, DTEF, DKR, DXLG, DTE, FKR, FKR1, PV
< REAL*8 TS(MXMEM), VA(MXMEM,MXNVI), WF
---
> REAL DIS, DSMX, DTEF, DKR, DXLG, DTE, FKR, FKR1, PV
> REAL TS(MXMEM), VA(MXMEM,MXNVI), WF
1923c1923
< REAL*8 ABFX
---
> REAL ABFX
1926c1926
< REAL*8 DX(MXPETH), TMN(MXTSP), BF(NDBF)
---
> REAL DX(MXPETH), TMN(MXTSP), BF(NDBF)
1931c1931
< REAL*8 V(MXNVI,MXSUB,MXPETH), DXX, B(2*MXNVI, MXSUB, MXPETH),
---
> REAL V(MXNVI,MXSUB,MXPETH), DXX, B(2*MXNVI, MXSUB, MXPETH),
2055,2057c2055,2057
< REAL*8 LAMBDA
< REAL*8 RMNT, T1, T2, AA, AY, FMNT, DT1, DT2, DT3, DT4, RNT
< REAL*8 DTSIG, DEPLEN, TM, FCTR, FNT, SIGT, CM
---
> REAL LAMBDA
> REAL RMNT, T1, T2, AA, AY, FMNT, DT1, DT2, DT3, DT4, RNT
> REAL DTSIG, DEPLEN, TM, FCTR, FNT, SIGT, CM
2060,2061c2060,2061
< cc REAL*8 DXK(MXMEM), DTMIN, DTX, CNX, VDT(MXPETH)
< REAL*8 DXK(MXPETH), DTMIN, DTX, CNX, VDT(MXPETH)
---
> cc REAL DXK(MXMEM), DTMIN, DTX, CNX, VDT(MXPETH)
> REAL DXK(MXPETH), DTMIN, DTX, CNX, VDT(MXPETH)
2065c2065
< REAL*8 CROUT(MXMEM), VELISO(MXPETH,MXMEM), TRAVT(MXISO),
---
> REAL CROUT(MXMEM), VELISO(MXPETH,MXMEM), TRAVT(MXISO),
2073c2073
< REAL*8 TH(MXLEG), PATH(MXLEG), Q(MXLEG)
---
> REAL TH(MXLEG), PATH(MXLEG), Q(MXLEG)
2077c2077
< REAL*8 Y, DT, PORE(MXPETH)
---
> REAL Y, DT, PORE(MXPETH)
2084c2084
< REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM),
---
> REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM),
2089c2089
< REAL*8 VOL, QSC
---
> REAL VOL, QSC
2093c2093
< REAL*8 TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPETH),
---

```

```

> REAL TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
2097c2097
< REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
---
> REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
2102c2102
< REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
> REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
2615,2616c2615,2616
< REAL*8 LAMBDA
< REAL*8 V1, VDT, VDTK, DT, Y
---
> REAL LAMBDA
> REAL V1, VDT, VDTK, DT, Y
2619c2619
< REAL*8 TH(MXLEG), Q(MXLEG), PATH(MXLEG)
---
> REAL TH(MXLEG), Q(MXLEG), PATH(MXLEG)
2623c2623
< REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
> REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
2631c2631
< REAL*8 PORE(MXPTH)
---
> REAL PORE(MXPTH)
2662,2663c2662,2663
< REAL*8 LAMBDA, D, EX1, EX2, TDNX
< REAL*8 FCTR, SUM, ANS, SFMX, CTF, XM, VR, VP, DT, BMN,
---
> REAL LAMBDA, D, EX1, EX2, TDNX
> REAL FCTR, SUM, ANS, SFMX, CTF, XM, VR, VP, DT, BMN,
2667c2667
< REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
> REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
2675c2675
< REAL*8 PORE(MXPTH)
---
> REAL PORE(MXPTH)
2682c2682
< REAL*8 F(NDDF), RHO(MXGRD,MXMEM), RHONEW(MXGRD),
---
> REAL F(NDDF), RHO(MXGRD,MXMEM), RHONEW(MXGRD),
2686c2686
< REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
> REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
3059,3062c3059,3062
< REAL*8 B1, B2, T1, T2, DT, S1, S2
< REAL*8 D(*),E1,E2,DN1,DN2,R1,R2,SUM,P
< REAL*8 T11,T12,T21,T22,T11T,T12T,T21T,T22T
< REAL*8 ANS, CJ, D1, D2
---
> REAL B1, B2, T1, T2, DT, S1, S2
> REAL D(*),E1,E2,DN1,DN2,R1,R2,SUM,P
> REAL T11,T12,T21,T22,T11T,T12T,T21T,T22T
> REAL ANS, CJ, D1, D2
3119c3119
< REAL*8 FUNCTION PRP(D,B,V,VP,DL,N,T1,T2)
---
> REAL FUNCTION PRP(D,B,V,VP,DL,N,T1,T2)
3123,3124c3123,3124
< REAL*8 D(*),F1,F2,DENOM,DF,P1,P2,P3,X

```



```

<      REAL*8 PROD, ZZ, T1, T2, B(*), V, VP, DL
---
>      REAL D(*),F1,F2,DENOM,DF,P1,P2,P3,X
>      REAL PROD, ZZ, T1, T2, B(*), V, VP, DL
3162,3163c3162,3163
<      REAL*8 LAMBDA, DSV
<      REAL*8 DTLT, DTLT2, DTSM, SUM, TDP
---
>      REAL LAMBDA, DSV
>      REAL DTLT, DTLT2, DTSM, SUM, TDP
3165c3165
<      REAL*8 TSPFAC
---
>      REAL TSPFAC
3167c3167
<      REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
>      REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
3175c3175
<      REAL*8 PORE(MXPTH), DT, Y
---
>      REAL PORE(MXPTH), DT, Y
3182c3182
<      REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
---
>      REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
3187c3187
<      REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
>      REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
3194c3194
<      REAL*8 BFR(MXMEM), D(MXMEM), DS(MXMEM)
---
>      REAL BFR(MXMEM), D(MXMEM), DS(MXMEM)
3311c3311
<      REAL*8 FUNCTION TSPFAC(DT,D,B,LT,TDP)
---
>      REAL FUNCTION TSPFAC(DT,D,B,LT,TDP)
3322,3324c3322,3324
<      REAL*8 D(*),DC(5),DN(9),A,SUM,SUM1,PM,P
<      REAL*8 F2,DCL,R12,R23,R24,R35, TDP
<      REAL*8 SUMS, DT, B(*)
---
>      REAL D(*),DC(5),DN(9),A,SUM,SUM1,PM,P
>      REAL F2,DCL,R12,R23,R24,R35, TDP
>      REAL SUMS, DT, B(*)
3429c3429
<      REAL*8 C, AW, P
---
>      REAL C, AW, P
3431c3431
<      REAL*8 BCJ(MXJCT)
---
>      REAL BCJ(MXJCT)
3438c3438
<      REAL*8 PATH(MXLEG), Q(MXLEG), TH(MXLEG)
---
>      REAL PATH(MXLEG), Q(MXLEG), TH(MXLEG)
3442c3442
<      REAL*8 PORE(MXPTH), DT, Y
---
>      REAL PORE(MXPTH), DT, Y
3656c3656
<      REAL*8 C, AW, AM, SUM
---
```

```

> REAL C, AW, AM, SUM
3658c3658
< REAL*8 BCJ(MXJCT)
---
> REAL BCJ(MXJCT)
3730,3732c3730,3732
< REAL*8 MKSP, MKSQ, MKSPOR, MKSQSC, MKSQDI, MKSY, PDIS, TT,
< 1 MKSPDI
< REAL*8 C, AW, P, PDF
---
> REAL MKSP, MKSQ, MKSPOR, MKSQSC, MKSQDI, MKSY, PDIS, TT,
> 1 MKSPDI
> REAL C, AW, P, PDF
3734c3734
< REAL*8 BCJ(MXJCT)
---
> REAL BCJ(MXJCT)
3741c3741
< REAL*8 PATH(MXLEG), Q(MXLEG), TH(MXLEG)
---
> REAL PATH(MXLEG), Q(MXLEG), TH(MXLEG)
3745c3745
< REAL*8 PORE(MXPETH), DT, Y
---
> REAL PORE(MXPETH), DT, Y
3752c3752
< REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
> REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
3756c3756
< REAL*8 VOL, QSC
---
> REAL VOL, QSC
3760c3760
< REAL*8 TDTM(MXTDV), TDSAT(MXTDV,MXPETH), TDVEL(MXTDV,MXPETH),
---
> REAL TDTM(MXTDV), TDSAT(MXTDV,MXPETH), TDVEL(MXTDV,MXPETH),
3951c3951
< REAL*8 C, AW
---
> REAL C, AW
3955c3955
< REAL*8 BCJ(MXJCT)
---
> REAL BCJ(MXJCT)
3960c3960
< REAL*8 TH(MXLEG), PATH(MXLEG), Q(MXLEG)
---
> REAL TH(MXLEG), PATH(MXLEG), Q(MXLEG)
3964c3964
< REAL*8 PORE(MXPETH), DT, Y
---
> REAL PORE(MXPETH), DT, Y
4017c4017
< REAL*8 T1, T2, DT, TUP, Y, DLG, DLT, DIS, DISI, PDIS,
---
> REAL T1, T2, DT, TUP, Y, DLG, DLT, DIS, DISI, PDIS,
4020c4020
< REAL*8 PATH(MXLEG), Q(MXLEG), TH(MXLEG)
---
> REAL PATH(MXLEG), Q(MXLEG), TH(MXLEG)
4024c4024
< REAL*8 PORE(MXPETH)
---
> REAL PORE(MXPETH)

```

```

4031c4031
< REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
> REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
4038c4038
< REAL*8 TDTM(MXTDV), TDSAT(MXTDV,MXPETH), TDVEL(MXTDV,MXPETH),
---
> REAL TDTM(MXTDV), TDSAT(MXTDV,MXPETH), TDVEL(MXTDV,MXPETH),
4115,4117c4115,4117
< REAL*8 DFF, EA, SUM, FMAX, T1, T2, TME, BLKS, TM1BLK
< REAL*8 TTRL, T3, DL, VR, VRP, TMAX, D(MXMEM), BFR(MXMEM)
< REAL*8 FM, TLED, TMIN, LAMBDA, DF1, RATF1
---
> REAL DFF, EA, SUM, FMAX, T1, T2, TME, BLKS, TM1BLK
> REAL TTRL, T3, DL, VR, VRP, TMAX, D(MXMEM), BFR(MXMEM)
> REAL FM, TLED, TMIN, LAMBDA, DF1, RATF1
4119c4119
< REAL*8 ET, PRP
---
> REAL ET, PRP
4121c4121
< REAL*8 CUROUT(MXMEM), VELISO(MXPETH,MXMEM), TRAVT(MXISO),
---
> REAL CUROUT(MXMEM), VELISO(MXPETH,MXMEM), TRAVT(MXISO),
4129c4129
< REAL*8 PORE(MXPETH), DT, Y
---
> REAL PORE(MXPETH), DT, Y
4136c4136
< REAL*8 DX(MXPETH), TMN(MXTSP), BF(NDBF)
---
> REAL DX(MXPETH), TMN(MXTSP), BF(NDBF)
4141c4141
< REAL*8 F(NDDF), SF(NDSF), RHO(MXGRD,MXMEM), RHONEW(MXGRD),
---
> REAL F(NDDF), SF(NDSF), RHO(MXGRD,MXMEM), RHONEW(MXGRD),
4145c4145
< REAL*8 V(MXNVI,MXSUB,MXPETH), DXX, B(2*MXNVI, MXSUB, MXPETH),
---
> REAL V(MXNVI,MXSUB,MXPETH), DXX, B(2*MXNVI, MXSUB, MXPETH),
4173,4176c4173,4175
< c rwj 9/18/97 Add line to prevent 0 subscript in RHO array for last member
< c in a chain of 3 or more.
< IF (L .GT. NTX) L = NTX
< c
---
> crbc92297
> if(l.gt.ntx) l=ntx
> crbc
4391,4392c4390,4391
< REAL*8 MKSPETH, MKSARE, MKSCON
< REAL*8 LAMBDA, C, AW, P
---
> REAL MKSPETH, MKSARE, MKSCON
> REAL LAMBDA, C, AW, P
4394c4393
< REAL*8 TIMBRK(MXISO), TIMPK(MXISO), PEAK(MXISO), WDIS(MXISO),
---
> REAL TIMBRK(MXISO), TIMPK(MXISO), PEAK(MXISO), WDIS(MXISO),
4399c4398
< REAL*8 BCJ(MXJCT)
---
> REAL BCJ(MXJCT)
4404c4403
< REAL*8 CUROUT(MXMEM), VELISO(MXPETH,MXMEM), TRAVT(MXISO),

```

```

---
> REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
4414c4413
< REAL*8 TH(MXLEG), PATH(MXLEG), Q(MXLEG)
---
> REAL TH(MXLEG), PATH(MXLEG), Q(MXLEG)
4418c4417
< REAL*8 PORE(MXPTH), DT, Y
---
> REAL PORE(MXPTH), DT, Y
4425c4424
< REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
> REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
4429c4428
< REAL*8 TSRCE(MXQSC), QSRCE(MXQSC)
---
> REAL TSRCE(MXQSC), QSRCE(MXQSC)
4435c4434
< REAL*8 VOL, QSC
---
> REAL VOL, QSC
4439c4438
< REAL*8 TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
---
> REAL TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
4443c4442
< REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
---
> REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
5311,5312c5310,5311
< REAL*8 LAMBDA, DTSAVE, PRF
< REAL*8 SOLG, FLCH, T, TDT, SUMA, SUMC, SUMU, TRAT, TW, TOTG,
---
> REAL LAMBDA, DTSAVE, PRF
> REAL SOLG, FLCH, T, TDT, SUMA, SUMC, SUMU, TRAT, TW, TOTG,
5320c5319
< REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
> REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
5328c5327
< REAL*8 PORE(MXPTH), DT, Y
---
> REAL PORE(MXPTH), DT, Y
5335c5334
< REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
> REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
5339c5338
< REAL*8 TSRCE(MXQSC), QSRCE(MXQSC)
---
> REAL TSRCE(MXQSC), QSRCE(MXQSC)
5342c5341
< REAL*8 VOL, QSC
---
> REAL VOL, QSC
5346c5345
< REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
> REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
5353c5352
< REAL*8 UNT(MXISO), AT(MXISO), PF(MXSUB,MXCHNS)
---
> REAL UNT(MXISO), AT(MXISO), PF(MXSUB,MXCHNS)
5775c5774

```

```

<      REAL*8 LAMBDA, T1, T2, TT, PDIS
---
>      REAL LAMBDA, T1, T2, TT, PDIS
5780c5779
<      REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
>      REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
5788c5787
<      REAL*8 Q(MXLEG), PATH(MXLEG), TH(MXLEG)
---
>      REAL Q(MXLEG), PATH(MXLEG), TH(MXLEG)
5792c5791
<      REAL*8 PORE(MXPTH), DT, Y
---
>      REAL PORE(MXPTH), DT, Y
5799c5798
<      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
>      REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
5806c5805
<      REAL*8 TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
---
>      REAL TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
5810c5809
<      REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
---
>      REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
5815c5814
<      REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
>      REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
5822c5821
<      REAL*8 VDT(MXPTH), DXSV(MXPTH), VDTSV(MXPTH)
---
>      REAL VDT(MXPTH), DXSV(MXPTH), VDTSV(MXPTH)
6047c6046
<      REAL*8 LAMBDA, ET
---
>      REAL LAMBDA, ET
6049c6048
<      REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
>      REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
6057c6056
<      REAL*8 PORE(MXPTH), DT, Y
---
>      REAL PORE(MXPTH), DT, Y
6062c6061
<      REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
>      REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
6077,6078c6076,6077
<      REAL*8 RM(MXMEM), DMIN, RMX, BFR(MXMEM)
<      REAL*8 D(MXMEM)
---
>      REAL RM(MXMEM), DMIN, RMX, BFR(MXMEM)
>      REAL D(MXMEM)
6282c6281
<      REAL*8 FUNCTION ET(T,IP,D,B)
---
>      REAL FUNCTION ET(T,IP,D,B)
6284,6285c6283,6284
<      REAL*8 D(*), DC(5), DN(5)
<      REAL*8 PROD, PM, F2, A, SUM, SUM1, T, B(*)
---

```

38/45

```
> REAL D(*), DC(5), DN(5)
> REAL PROD, PM, F2, A, SUM, SUM1, T, B(*)
6387,6388c6386,6387
< REAL*8 TIME, DJ, DI, RAT
< REAL*8 T,A1,A2,ARG1,ARG2,A,ARG,ELF,ADD,S,TS,TSS
---
> REAL TIME, DJ, DI, RAT
> REAL T,A1,A2,ARG1,ARG2,A,ARG,ELF,ADD,S,TS,TSS
6491c6490
< REAL*8 RESIDE, DT1, DT2, DT3, DT4, DTH, TTMIN, QSKQ, QMAX,
---
> REAL RESIDE, DT1, DT2, DT3, DT4, DTH, TTMIN, QSKQ, QMAX,
6493c6492
< REAL*8 LAMBDA, D, PROD, DEC
---
> REAL LAMBDA, D, PROD, DEC
6495c6494
< REAL*8 ET
---
> REAL ET
6497c6496
< REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
> REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
6505c6504
< REAL*8 PORE(MXPTH), DT, Y
---
> REAL PORE(MXPTH), DT, Y
6512c6511
< REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
> REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
6516c6515
< REAL*8 TSRCE(MXQSC), QSRCE(MXQSC)
---
> REAL TSRCE(MXQSC), QSRCE(MXQSC)
6519c6518
< REAL*8 QSC, VOL
---
> REAL QSC, VOL
6523c6522
< REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
> REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
6530c6529
< REAL*8 BFR(MXISO), AT(MXISO), PF(MXSUB,MXCHNS)
---
> REAL BFR(MXISO), AT(MXISO), PF(MXSUB,MXCHNS)
6774c6773
< REAL*8 QSC, QDIS, VOL, T1, T2, F, F1, TSTOP, ARG
---
> REAL QSC, QDIS, VOL, T1, T2, F, F1, TSTOP, ARG
6776c6775
< REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM)
---
> REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM)
6825c6824
< REAL*8 LAMBDA
---
> REAL LAMBDA
6838c6837
< REAL*8 TS, T1, TRAT, DELT, T
---
> REAL TS, T1, TRAT, DELT, T
6842c6841
```

```

<      REAL*8 PEAK(MXISO), TOTD(MXISO), WDIS(MXISO), WDISUM,
---
>      REAL PEAK(MXISO), TOTD(MXISO), WDIS(MXISO), WDISUM,
6850c6849
<      REAL*8 CROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
>      REAL CROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
6860c6859
<      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
>      REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
6865c6864
<      REAL*8 TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
---
>      REAL TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
6870c6869
<      REAL*8 TSCL(NTMX), RATE(MXISO), RATEL(MXISO), RATER(MXISO),
---
>      REAL TSCL(NTMX), RATE(MXISO), RATEL(MXISO), RATER(MXISO),
7191c7190
<      REAL*8 X(*), A
---
>      REAL X(*), A
7218c7217
<      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
>      REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
7260c7259
<      REAL*8 PORE(MXPTH), DT, Y
---
>      REAL PORE(MXPTH), DT, Y
7265c7264
<      REAL*8 QSC, VOL
---
>      REAL QSC, VOL
7395,7397c7394,7396
<      REAL*8 LAMBDA, C, AW, P
<      REAL*8 MKSQSA, MKSSAL, MKSQDA, MKSARE, MKSALP, MKSDX, MKSP
<      REAL*8 MKSCON, MKSPOR, MKSEL, MKSTDV, MKSVOL, MKSPTH, MKSQSR
---
>      REAL LAMBDA, C, AW, P
>      REAL MKSQSA, MKSSAL, MKSQDA, MKSARE, MKSALP, MKSDX, MKSP
>      REAL MKSCON, MKSPOR, MKSEL, MKSTDV, MKSVOL, MKSPTH, MKSQSR
7399,7400c7398,7399
<      REAL*8 PEAK(MXISO), TIMBRK(MXISO), TIMPK(MXISO), TOTD(MXISO)
<      REAL*8 WDIS(MXISO), WDISUM
---
>      REAL PEAK(MXISO), TIMBRK(MXISO), TIMPK(MXISO), TOTD(MXISO)
>      REAL WDIS(MXISO), WDISUM
7404c7403
<      REAL*8 BCJ(MXJCT)
---
>      REAL BCJ(MXJCT)
7412c7411
<      REAL*8 CROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
>      REAL CROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
7422c7421
<      REAL*8 TH(MXLEG), Q(MXLEG), PATH(MXLEG)
---
>      REAL TH(MXLEG), Q(MXLEG), PATH(MXLEG)
7426c7425
<      REAL*8 PORE(MXPTH), DT, Y
---
>      REAL PORE(MXPTH), DT, Y

```

40/45

```

7433c7432
< REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
> REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
7437c7436
< REAL*8 TSRCE(MXQSC), QSRCE(MXQSC)
---
> REAL TSRCE(MXQSC), QSRCE(MXQSC)
7443c7442
< REAL*8 QSC, VOL
---
> REAL QSC, VOL
7447c7446
< REAL*8 TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
---
> REAL TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
7451c7450
< REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
---
> REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
7456c7455
< REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
> REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
7774,7775c7773,7774
< REAL*8 CHK, TM, T, SIGT, TL, VV, TSTART, TEND, T1, T2, TSUM
< REAL*8 LAMBDA
---
> REAL CHK, TM, T, SIGT, TL, VV, TSTART, TEND, T1, T2, TSUM
> REAL LAMBDA
7777c7776
< REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
> REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
7785c7784
< REAL*8 PORE(MXPTH), DT, Y
---
> REAL PORE(MXPTH), DT, Y
7792c7791
< REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
> REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
7799c7798
< REAL*8 QSC, VOL
---
> REAL QSC, VOL
7803c7802
< REAL*8 TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
---
> REAL TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
7807c7806
< REAL*8 F(NDDF), RHOM(MXGRD,MXMEM), RHONEW(MXGRD), SF(NDSF),
---
> REAL F(NDDF), RHOM(MXGRD,MXMEM), RHONEW(MXGRD), SF(NDSF),
7818,7821c7817,7820
< REAL*8 curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
< REAL*8 dr2tim(maxrec), dr2vol(maxrec)
< REAL*8 filcur(mxiso,maxrec), filtim(maxrec)
< REAL*8 filvol(maxrec), tarr(maxtim), totvol(maxtim)
---
> REAL curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
> REAL dr2tim(maxrec), dr2vol(maxrec)
> REAL filcur(mxiso,maxrec), filtim(maxrec)
> REAL filvol(maxrec), tarr(maxtim), totvol(maxtim)
7984,7985c7983,7984

```



```

<      REAL*8 LAMBDA, RAT12, RAT23, RAT13, GIT
<      REAL*8 E1, E2, E3, B, C, GPART, A, V, T, TL, TSET,
---
>      REAL LAMBDA, RAT12, RAT23, RAT13, GIT
>      REAL E1, E2, E3, B, C, GPART, A, V, T, TL, TSET,
7994c7993
<      REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
>      REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
8002c8001
<      REAL*8 PORE(MXPTH), DT, Y
---
>      REAL PORE(MXPTH), DT, Y
8009c8008
<      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
>      REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
8013c8012
<      REAL*8 QSC, VOL
---
>      REAL QSC, VOL
8083c8082
<      REAL*8 FUNCTION GIT(T,X,V,AL)
---
>      REAL FUNCTION GIT(T,X,V,AL)
8085,8086c8084,8085
<      REAL*8 A(5), P, TRPI, T, X, V, AL, ANS, B, Z
<      REAL*8 PLYET1, PLYET2, ET1, ET2, ZMB, TERM1, TERM2, XX, XXEX
---
>      REAL A(5), P, TRPI, T, X, V, AL, ANS, B, Z
>      REAL PLYET1, PLYET2, ET1, ET2, ZMB, TERM1, TERM2, XX, XXEX
8145c8144
<      REAL*8 T, MKSQDI, DT2, DTFP , LAMBDA, TM, TSET
---
>      REAL T, MKSQDI, DT2, DTFP , LAMBDA, TM, TSET
8147,8150c8146,8149
<      REAL*8 totvol(maxtim)
<      REAL*8 curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
<      REAL*8 dr2tim(maxrec), dr2vol(maxrec), filtim(maxrec)
<      REAL*8 filcur(mxiso,maxrec), filvol(maxrec), tarr(maxtim)
---
>      REAL totvol(maxtim)
>      REAL curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
>      REAL dr2tim(maxrec), dr2vol(maxrec), filtim(maxrec)
>      REAL filcur(mxiso,maxrec), filvol(maxrec), tarr(maxtim)
8154c8153
<      REAL*8 PEAK(MXISO), TODD(MXISO), WDIS(MXISO), WDISUM,
---
>      REAL PEAK(MXISO), TODD(MXISO), WDIS(MXISO), WDISUM,
8159c8158
<      REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
>      REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
8167c8166
<      REAL*8 PORE(MXPTH), DT, Y
---
>      REAL PORE(MXPTH), DT, Y
8174c8173
<      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
>      REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
8279,8280c8278,8279
<      REAL*8 LAMBDA, T, SUM, SUM1, SUM2, SUM3, SUM4, SUM5, SUM6
<      REAL*8 TX1, TATM, TX2, SR1, SR2, TS1, TS2, S24, TSET, BM
---
```

```

>      REAL LAMBDA, T, SUM, SUM1, SUM2, SUM3, SUM4, SUM5, SUM6
>      REAL TX1, TATM, TX2, SR1, SR2, TS1, TS2, S24, TSET, BM
8282c8281
<      REAL*8 ABFX
---
>      REAL ABFX
8288c8287
<      REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
>      REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
8296c8295
<      REAL*8 PORE(MXPTH), DT, Y
---
>      REAL PORE(MXPTH), DT, Y
8303c8302
<      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
>      REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
8307c8306
<      REAL*8 QSC, VOL
---
>      REAL QSC, VOL
8311c8310
<      REAL*8 TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
---
>      REAL TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
8315c8314
<      REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
---
>      REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
8320c8319
<      REAL*8 F(NDDF), SF(NDSF), RHO(MXGRD,MXMEM), RHONEW(MXGRD),
---
>      REAL F(NDDF), SF(NDSF), RHO(MXGRD,MXMEM), RHONEW(MXGRD),
8324c8323
<      REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
>      REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
8575,8576c8574,8575
<      REAL*8 LAMBDA, TX1, TX2, TS1, TS2, SR1, SR2, S24
<      REAL*8 S, SUM, TP, TRAT, TRAT2, RT, RT2
---
>      REAL LAMBDA, TX1, TX2, TS1, TS2, SR1, SR2, S24
>      REAL S, SUM, TP, TRAT, TRAT2, RT, RT2
8578c8577
<      REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
>      REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
8586c8585
<      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
>      REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
8590c8589
<      REAL*8 QSC, VOL
---
>      REAL QSC, VOL
8594c8593
<      REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
>      REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
8601c8600
<      REAL*8 F(NDDF), SF(NDSF), RHO(MXGRD,MXMEM), RHONEW(MXGRD),
---
>      REAL F(NDDF), SF(NDSF), RHO(MXGRD,MXMEM), RHONEW(MXGRD),
8743c8742

```

```

<      REAL*8 LAMBDA, CF, TA, PC, CM, DIF, DPI
---
>      REAL LAMBDA, CF, TA, PC, CM, DIF, DPI
8745c8744
<      REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
>      REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
8753c8752
<      REAL*8 PORE(MXPTH), DT, Y
---
>      REAL PORE(MXPTH), DT, Y
8761c8760
<      REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
---
>      REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
8765c8764
<      REAL*8 F(NDDF), SF(NDSF), RHO(MXGRD,MXMEM), RHOM(MXGRD,MXMEM),
---
>      REAL F(NDDF), SF(NDSF), RHO(MXGRD,MXMEM), RHOM(MXGRD,MXMEM),
8769c8768
<      REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
>      REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
8885c8884
<      REAL*8 dummyarray(maxtim,1), LAMBDA, tempc(mxiso)
---
>      REAL dummyarray(maxtim,1), LAMBDA, tempc(mxiso)
8887,8890c8886,8889
<      REAL*8 curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
<      REAL*8 dr2tim(maxrec), dr2vol(maxrec)
<      REAL*8 filcur(mxiso,maxrec), filtim(maxrec)
<      REAL*8 filvol(maxrec), tarr(maxtim), totvol(maxtim)
---
>      REAL curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
>      REAL dr2tim(maxrec), dr2vol(maxrec)
>      REAL filcur(mxiso,maxrec), filtim(maxrec)
>      REAL filvol(maxrec), tarr(maxtim), totvol(maxtim)
8895c8894
<      REAL*8 CUROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
---
>      REAL CUROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
8903c8902
<      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
>      REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
8959c8958
<      REAL*8 dummyarray(maxtim,1), LAMBDA, tempc(mxiso), tempv(maxtim)
---
>      REAL dummyarray(maxtim,1), LAMBDA, tempc(mxiso), tempv(maxtim)
8961c8960
<      REAL*8 curies(maxtim,mxiso), dr2tim(maxrec), filvol(maxrec),
---
>      REAL curies(maxtim,mxiso), dr2tim(maxrec), filvol(maxrec),
8967c8966
<      REAL*8 CUROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
---
>      REAL CUROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
8975c8974
<      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
>      REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
9011c9010
<      REAL*8 LAMBDA
---
>      REAL LAMBDA

```

```

9013,9016c9012,9015
< REAL*8 curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
< REAL*8 dr2tim(maxrec), dr2vol(maxrec)
< REAL*8 filcur(mxiso,maxrec), filtim(maxrec)
< REAL*8 filvol(maxrec), tarr(maxtim), totvol(maxtim)
---
> REAL curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
> REAL dr2tim(maxrec), dr2vol(maxrec)
> REAL filcur(mxiso,maxrec), filtim(maxrec)
> REAL filvol(maxrec), tarr(maxtim), totvol(maxtim)
9021c9020
< REAL*8 CROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
---
> REAL CROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
9029c9028
< REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
> REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
9067c9066
< real*8 slope, t(maxtim), delt, dt, var(maxtim,nmax), val(nmax)
---
> real slope, t(maxtim), delt, dt, var(maxtim,nmax), val(nmax)
9114c9113
< REAL*8 LAMBDA
---
> REAL LAMBDA
9123c9122
< REAL*8 conc, u234(maxrec), u234tm(maxrec)
---
> REAL conc, u234(maxrec), u234tm(maxrec)
9125,9128c9124,9127
< REAL*8 curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
< REAL*8 dr2tim(maxrec), dr2vol(maxrec)
< REAL*8 filcur(mxiso,maxrec), filtim(maxrec)
< REAL*8 filvol(maxrec), tarr(maxtim), totvol(maxtim)
---
> REAL curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
> REAL dr2tim(maxrec), dr2vol(maxrec)
> REAL filcur(mxiso,maxrec), filtim(maxrec)
> REAL filvol(maxrec), tarr(maxtim), totvol(maxtim)
9136c9135
< REAL*8 CROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
---
> REAL CROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
9210c9209
< REAL*8 LAMBDA
---
> REAL LAMBDA
9219c9218
< REAL*8 conc
---
> REAL conc
9221c9220
< REAL*8 CROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
---
> REAL CROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
9232c9231
< REAL*8 curies(maxtim,mxiso), dr2cur(mxiso,maxrec), tarr(maxtim),
---
> REAL curies(maxtim,mxiso), dr2cur(mxiso,maxrec), tarr(maxtim),

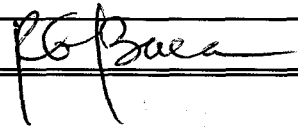
```

TPA V 3.1.2


SOFTWARE RELEASE NOTICE

| | | |
|--|----------------------|--------------------------|
| 1. SRN Number: PA-SRN-207 | | |
| 2. Project Title: TSPA & Technical Integration Code | | Project No. 20-01402-762 |
| 3. SRN Title: TPA Version 3.1.2 hf 10-4-99 | | |
| 4. Originator/Requestor: Ron Janetzke | | Date: Oct. 1, 1999 |
| 5. Summary of Actions <ul style="list-style-type: none"> <input type="checkbox"/> Release of new software <input type="checkbox"/> Release of modified software: <ul style="list-style-type: none"> <input type="checkbox"/> Enhancements made <input type="checkbox"/> Corrections made <input type="checkbox"/> Change of access software <input checked="" type="checkbox"/> Software Retirement | | |
| 6. Persons Authorized Access | | |
| Name | Read Only/Read-Write | Addition/Change/Delete |
| Ron Janetzke | RW | |
| Sitakanta Mohanty | RW | |
| Tim McCartin (NRC) | RW | |
| M. Rose Byrne (NRC) | RW | |
| 7. Element Manager Approval: <i>Arden Wittmeyer</i> | | Date: 10/4/99 |
| 8. Remarks: Version superceded. | | |

SOFTWARE RELEASE NOTICE

| | | |
|---|--|-----------------------------|
| 01. SRN Number: PA-SRN-164 | | |
| 02. Project Title: TSPA & Technical Integration Code | | Project No.: 20-1402-762 |
| 03. SRN Title: TPA Version 3.1.2 | | |
| 04. Originator/Requestor: Bruce Mabrito | | Date: 1/16/98 |
| 05. Summary of Actions <input type="checkbox"/> Release of new software <input checked="" type="checkbox"/> Release of modified software: <input checked="" type="checkbox"/> Enhancements made <input type="checkbox"/> Corrections made <input type="checkbox"/> Change of access software <input type="checkbox"/> Software Retirement | | |
| 06. Persons Authorized Access | | |
| Name | RO/RW | A/C/D |
| Sitakanta Mohanty Ron Janetzke Tim McCartin M. Rose Byrne | READ/WRITE READ/WRITE READ/WRITE READ/WRITE | |
| 07. Element Manager Approval:  | | Date: 1/16/98 |
| 08. Remarks: TPA Version 3.1.2 software was sent to NRC Headquarters on 01/16/98. | | |

SOFTWARE SUMMARY FORM

| | | | |
|--|---|---|--|
| 01. Summary Date: 01/16/98 | 02. Summary prepared by (Name and phone) B. Mabrito (210)522-5149 | 03. Summary Action: Modified | |
| 04. Software Date: 01/16/98 | 05. Short Title: TPA Version 3.1.2 | | |
| 06. Software Title: TPA — Total-System Performance Assessment Computer Code, Version 3.1.2 | | 07. Internal Software ID: None | |
| 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 type="checkbox"/> Scientific/Engineering <input type="checkbox"/> Auxiliary Analyses <input checked="" type="checkbox"/> Total System PA <input type="checkbox"/> Subsystem PA <input type="checkbox"/> Other B. Specific: | |
| 11. Submitting Organization and Address: CNWRA/SwRI 6220 Culebra Road San Antonio, TX 78238 | | 12. Technical Contact(s) and Phone: Sitakanta Mohanty (210)522-5185 | |
| 13. Narrative: The TPA Code consists of the following modules: UZFLOW, NFENV, EBSREL, UZFT, SZFT, DCAGW, FAULTO, SEISMO, VOLCANO, ASHPLUMO, ASHRMVO, DCAGS, LHS, EXEC. | | | |
| 14. Computer Platform: SUN Workstation | 15. Computer Operating System: UNIX | 16. Programming Language(s): FORTRAN | 17. Number of Source Program Statements: approx. 32,000 lines |
| 18. Computer Memory Requirements: Unknown | 19. Tape Drives: None | 20. Disk/Drum Units: N/A | 21. Graphics: N/A |
| 22. Other Operational Requierments: None | | | |
| 23. Software Availability: <input checked="" type="checkbox"/> Available <input type="checkbox"/> Limited <input type="checkbox"/> In-House ONLY | | 24. Documentation Availability: <input type="checkbox"/> Available <input type="checkbox"/> Inadequate <input checked="" type="checkbox"/> In-House ONLY | |
| Software Custodian:  | | Date: 1/16/98 | |

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE: TPA Version 3.1.2

January 16, 1998

Total Performance Assessment (Scientific and Engineering Software) Version 3.1.2

1. Scientific Notebook Documentation Development: CNWRA Scientific Notebooks numbered: 170 - S. Mohanty and R. Janetzke (EBSFAIL, EBSREL, UGFT, UZFT, and SZFT); 202 - S. Mohanty (EXEC, SEISMO, DCAGS, DCAGW, VOLCANO, NFENV, and UTILITY modules). These scientific notebooks document the individual software development, along with the 19 attached Software Problem/Change Reports (SPCRs) that were generated by both the CNWRA and NRC staff working on TPA Version 3.1.2.
2. Programming Language: ANSI Standard FORTRAN 77 confirmed by the Software Custodian. See attached hard copies of TPA Version 3.1.2 pages as examples.
3. Internal Documentation: On 1/16/98, B. Mabrito reviewed portions of the TPA Version 3.1.2 scientific and engineering software on the "Bigbend" SUN Ultra 2 work station. Comment lines in the subroutines were observed by the software custodian. The TPA Version 3.1.2 software is located on the "Scratchy1" server. The TPA code contained comments and a hard copy page showing representative comments is attached to this Design Verification Report. The internal documentation comments in TPA Version 3.1.2 meet the requirements of TOP-018, Section 5.4.4.
4. Software Labels and Data
 - a. Header Data and Format: TPA Version 3.1.2 header data and the format were compared against TOP-018 Section 5.4.6 and found acceptable (see several of the attached hard copy pages).
 - b. NRC Data: TPA Version 3.1.2 NRC data and the format were compared against TOP-018, Section 5.4.6, third bullet and found to be acceptable.
 - c. Source Code Header: TPA Version 3.1.2 header data was compared to TOP-018 Section 5.4.6, fourth bullet, and found to meet the requirements and are acceptable.
5. Unique Run Identification: At the top of each screen output file for TPA Version 3.1.2, a unique identifier is on the printed sheet. A page of files was printed and attached to this report to show the following statement: "Welcome to TPA 3.1.2, Job started: Fri Jan 16 19:32:05 1998" which fully meets the unique run identification requirements of TOP-018, Section 5.4.5.

6. Software Analysis and Results

a. Analysis: FOR_STUDY Version 1.2 was utilized as the software analysis tool for developing TPA Version 3.1.2. FOR_STUDY was run in the "default configuration" as specified in the Users Manual and run on all modules. Documentation was captured from such runs by C. Scherer and some of it is attached to this report.

b. Analysis Report: Portions of one of the final analysis reports has been printed and numerous pages have been attached to this document. C. Scherer answered questions of the CNWRA Code Custodian regarding use of FOR_STUDY. FOR_STUDY was utilized to debug the software and results of the FOR_STUDY runs were conveyed to the Software Developer. A FOR_STUDY run was performed on each of the final versions of the code. The Software Developer authorized such FOR_STUDY runs and was cognizant of all such testing of TPA Version 3.1.2.

c. Resolution of Comments: The TPA Version 3.1.2 Software Developer was made aware of the warning and error messages by C. Scherer. According to the Software Developer, the warning and error messages from the FOR_STUDY runs were evaluated but no changes were necessary to reach the final TPA Version 3.1.2. There were a few non fatal error messages, but they were determined not to affect the results and were accepted by the Software Developer.



1/16/98

Sitakanta Mohanty
CNWRA TPA Software Developer



1/16/98

Bruce Mabrito
CNWRA Software Custodian

Attachments/

original to: Software Folder
cc: CNWRA Software Developer

3/14

=====

```
exec: Welcome to Version 3.1.2
Job started: Fri Jan 16 19:32:05 1998
```

=====

Specified Global Parameters:

```
NumberOfRealizations = 1
NumberOfSubareas = 7
MaximumTime[yr] = 10000.
Volcanism scenario = 0 (yes=1, no=0)
Faulting scenario = 0 (yes=1, no=0)
Seismic scenario = 0 (yes=1, no=0)
Distance to Critical Group (km) = 20.0
```

*Unique
Identification
SS
1/16/98*

```
The specified path for data = /bscr3/mohanty/A_tpa3.1.2/
The specified path for codes = /bscr3/mohanty/A_tpa3.1.2/
```

****To modify global parameters or the path, stop code execution using control-C****

```
subarea 1 of 7 realization 1 of 1
```

```
exec: calling uzflow
exec: calling nfenv
exec: calling ebsfail
      ebsfail: time of WP failure = 4092.7 yr
exec: failed WPs from INITIAL event = 13 at time = 0.0 yr
exec: failed WPs from CORROSION event = 1650 at time = 4092.7 yr
      *** failed WPs: all WPs failed ***
exec: calling ebsrel
```

Input file tpa.inp as supplied with TPA Version 3.1 Code.

Base case data set Rev 3.1.1 1/15/98

TPA 3.1.2, Job started: Fri Jan 16 19:32:58 1998 ✓

Total Dose for All Pathways, All Nuclides, and All Times


4/14

Including Dilution Volume - Values for Each Vector

| vector unitless | time yr | tede rem/yr | dilutvol m ³ /yr |
|--------------------|------------|----------------|--------------------------------|
| 1 | 0.0000E+00 | 0.0000E+00 | 1.2680E+07 |
| 1 | 2.3102E+00 | 0.0000E+00 | 1.2680E+07 |
| 1 | 4.6744E+00 | 0.0000E+00 | 1.2680E+07 |
| 1 | 7.0940E+00 | 0.0000E+00 | 1.2680E+07 |
| 1 | 9.5702E+00 | 0.0000E+00 | 1.2680E+07 |
| 1 | 1.2104E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 1.4698E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 1.7352E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 2.0069E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 2.2849E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 2.5694E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 2.8605E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 3.1585E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 3.4635E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 3.7756E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 4.0950E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 4.4219E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 4.7564E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 5.0988E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 5.4492E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 5.8078E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 6.1747E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 6.5503E+01 | 0.0000E+00 | 1.2680E+07 |
| 1 | 6.9347E+01 | 0.0000E+00 | 1.2680E+07 |

5/14

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

Date: January 16, 1998
To: Tim McCartin
From: Sitakanta Mohanty 
Subject: Delivery of TPA code Version 3.1.2

Enclosed please find a copy of TPA code Version 3.1.2 in an 8-mm tape. TPA code version 3.1.2 is an updated version of TPA code version 3.1.1 and includes the following major changes suggested during a series of NRC/CNWRA topic-specific meetings.

- SZFT was modified to account for matrix diffusion in the saturated zone.
- The stand-alone module *releaset.f* was modified to add flow-through model (in addition to bath tub model) for spent fuel leaching.
- DCAGS was modified to use mass loading factor instead of a resuspension factor.
- DCAGW was modified to capture a fraction of the plume at 5 km instead of the whole plume as was done before. Also, the dilution volume is now always the well pumping rate at 5 km.
- UZFT was modified to generate and read the NEFTRAN discharge file to avoid errors associated with round off of time while reading *nefti.out* file.
- EXEC was modified to provide an option to the user to select a specified range of realizations and subareas in *tpa.inp*.

cc: K. McConnell, R. Baca, B. Sagar, B. Mabrito, R. Janetzke

Memo sent with
TPA Version 3.1.2 Tape
To NRC - BSW
1/16/98

6/14

tpa312/CLEANUP
tpa312/Makefile
tpa312/README
tpa312/array.f
tpa312/ashplumo.f
tpa312/ashrmovo.f
tpa312/conduxyzt.f
tpa312/dcags.f
tpa312/dcagw.f
tpa312/ebsfail.f
tpa312/ebsrel.f
tpa312/exec.f
tpa312/execa.i
tpa312/execb.i
tpa312/faulto.f
tpa312/fileunit.f
tpa312/findelev.f
tpa312/fu1.i
tpa312/fu2.i
tpa312/invent.f
tpa312/inventa.i
tpa312/inventb.i
tpa312/inventc.i
tpa312/inventd.i
tpa312/invente.i
tpa312/inventf.i
tpa312/inventg.i
tpa312/inventh.i
tpa312/inventi.i
tpa312/inventj.i
tpa312/inventk.i
tpa312/inventl.i
tpa312/inventm.i
tpa312/inventn.i
tpa312/max500yr.i
tpa312/maxchain.i
tpa312/maxnnucl.i
tpa312/maxnsuba.i
tpa312/maxntime.i
tpa312/mv.f
tpa312/mva.i
tpa312/mvb.i
tpa312/mvc.i
tpa312/mvd.i
tpa312/mve.i
tpa312/mvf.i
tpa312/nfenv.f
tpa312/nintv.i
tpa312/notice.i
tpa312/numrecip.f
tpa312/path.i
tpa312/peakfinder.f
tpa312/ran.f
tpa312/reader.f
tpa312/reflux2.i
tpa312/sampler.f
tpa312/sampler0.i
tpa312/sampler1.i
tpa312/sampler2.i
tpa312/sampler.a.i
tpa312/samplerb.i
tpa312/samplerc.i
tpa312/samplerd.i
tpa312/sampler.e.i
tpa312/samplerf.i

File Names
on TPA Version
3.1.2 TAP E

BSM 1/16/98

7/14

```
tpa312/samplerg.i
tpa312/samplerh.i
tpa312/sampleri.i
tpa312/samplerj.i
tpa312/samplerk.i
tpa312/samplerl.i
tpa312/samplerm.i
tpa312/samplern.i
tpa312/samplero.i
tpa312/samplerp.i
tpa312/samplerq.i
tpa312/samlerr.i
tpa312/samplers.i
tpa312/samlert.i
tpa312/sampleru.i
tpa312/samlerv.i
tpa312/samplerw.i
tpa312/samlerox.i
tpa312/samlery.i
tpa312/samplerz.i
tpa312/seismo.f
tpa312/setfiles.i
tpa312/stop.i
tpa312/subarea.f
tpa312/subareaa.i
tpa312/subareab.i
tpa312/subareac.i
tpa312/subaread.i
tpa312/subareae.i
tpa312/szft.f
tpa312/szft.i
tpa312/tpa.inp
tpa312/tpa.inp.meanvalues
tpa312/uz_climi.i
tpa312/uz_climr.i
tpa312/uz_climz.i
tpa312/uz_flowi.i
tpa312/uz_flowr.i
tpa312/uz_flowz.i
tpa312/uz_parms.i
tpa312/uzflow.f
tpa312/uzft.f
tpa312/volcano.f
```

```
tpa312/ccdf:
Makefile
tccdf.f
tccdf.i
tccdf.inp
```

```
tpa312/codes:
Makefile
README
SIZES.INC
SIZES2.INC
ashplume.f
fault.f
nefmks.f
reaset.f
snllhs.f
```

```
tpa312/data:
climato1.dat
climato2.dat
dilution.dat
```

8/14

ebsfail.def
ebspac.nuc
ebsrel.def
elevdem.dat
gs_cb_ad.dat
gs_cb_ci.dat
gs_pb_ad.dat
gs_pb_ci.dat
gw_cb_ad.dat
gw_cb_ci.dat
gw_pb_ad.dat
gw_pb_ci.dat
multiflo.dat
rectedge.dat
seismo.dat
soildem.dat
strmtube.dat
tefkti.inp
tpanames.dbs

ls.out

Screen examples

10/14

```

c
    implicit double precision (a-h,o-z)
    implicit integer (i-n)

    include 'maxntime.i'
    include 'maxnsuba.i'
    include 'maxnnucl.i'
cc rwr 9/3/97 modified to remove for_study messages
cc    include 'setfiles.i'
    include 'execa.i'
    include 'execb.i'

    include 'inventb.i'
cc rwr insertcasl 6/9/97 Added for spfilter.e
    include 'path.i'

cc rwr 6/7/97 modifications for output

    include 'samplerv.i'
    include 'samplerb.i'
    include 'samplerd.i'

cc rwr end of modifications

cc rwj 12-17-97 Include files for call to climato for first subarea.
    include 'uz_parms.i'
    include 'uz_flowz.i'

csm8/11 changes for larger maxtim

csm8/11    parameter (maxseismicevents = 150 )
           parameter (maxseismicevents = 250 )

           character*6 names(maxnnucl)
           character*60 name

cc rwr 5/16/97 modified because included in setfiles.i
cc    character*160 mes1
cc    character*160 mes2
cc rwj 9-3-97 Add ieee error handler.
           character*80 ieeeout

           dimension tim(maxntime)
           dimension qm3peryrinsa( maxntime )
           dimension qm3peryrperwpinsa( maxntime )
           dimension temprep( maxntime )
           dimension tempwp( maxntime )
           dimension tempsf( maxntime )
           dimension relhumwp( maxntime )
           dimension phwp( maxntime )
           dimension clwp( maxntime )
           dimension qm3peryrperwpinsamisswp( maxntime )
           dimension qm3peryrperwpinsahitwp( maxntime )
cc rwj 4-16-97
           dimension cumfailwp( maxntime )
cc
           dimension corrosionfailwp( maxntime )
           dimension seismicfailwp( maxntime )
           dimension faultfailwp( maxntime, maxnsubarea )
           dimension volcanicfailwp( maxntime, maxnsubarea )
           dimension volcanicantufailed( maxnsubarea )
crwr dimension ciperyrperwpinsafromwp( maxntime, maxnnucl )
cc    dimension qm3peryrinsaintoloweruz( maxntime )
           dimension ciperyrinsaintoloweruz( maxntime, maxnnucl )
           dimension qm3peryrinsafromuz( maxntime )

```

Comments

11/14

```

dimension ciperyrinsafromuz( maxntime, maxnnucl )
cc-new
cc rwr 6/13/97 removed the following lines
cc      dimension qm3peryrinsafromsz( maxntime )
cc      dimension qm3peryrallsafromsz( maxntime )
dimension ciperyrinsafromsz( maxntime, maxnnucl )
dimension ciperyrallsafromsz( maxntime, maxnnucl )

dimension ciper2gsatCP(maxntime, 43 )
dimension remperyrwnr(maxntime, maxnnucl)
dimension remperyrwswr(maxntime)
dimension remperyrwna(maxntime,maxnnucl)
dimension remperyrwsa(maxntime)
dimension remperyrwnr(maxntime, 43 )
dimension remperyrwssr(maxntime)
dimension remperyrwna(maxntime, 43 )
dimension remperyrwssa(maxntime)
dimension remperyrwgssa(maxntime)

cc
cc      rgwnr.tpa      <- remperyrwnr(maxntime,maxnnucl)
cc      rgwswr.tpa     <- remperyrwswr(maxntime)
cc      rgwna.tpa      <- remperyrwna(maxntime)
cc      rgwsa.tpa      <- remperyrwsa(maxntime)
cc
cc      rgsnr.tpa      <- remperyrwnr(maxntime,maxnnucl)
cc      rgssr.tpa      <- remperyrwssr(maxntime)
cc      rgsna.tpa      <- remperyrwna(maxntime)
cc      rgssa.tpa      <- remperyrwssa(maxntime)
cc
cc      rgwgssa.tpa    <- remperyrwgssa(maxntime)
cc
dimension timeofseismicevents(maxseismicevents)
dimension typeofseismicevents(maxseismicevents)

dimension ipt(43)
dimension temp(43)

csm8/13      double precision maxrel( maxnnucl )
csm8/13      double precision maxreltim( maxnnucl )
double precision maxrel( 50 )
double precision maxreltim( 50 )

cc rwr 6/7/97 modifications for output
parameter (maxrealizations = 500 )
dimension gwdosemax(maxnnucl,maxrealizations)
dimension gwtimemaxdose(maxnnucl,maxrealizations)

dimension remperyrwnsr(maxntime,43)

dimension ntimemax(maxrealizations)
dimension totalmaxdosetime(maxrealizations)
dimension totalmaxdose(maxrealizations)
dimension gsnmaxdose(maxrealizations,43)

dimension gwt( maxsubarea, 2)

dimension gwtttotuzsz(maxsubarea)

dimension cumrelease(maxnnucl,maxrealizations,maxsubarea,3)

dimension peakrelrate(maxnnucl,maxrealizations,maxsubarea)
dimension peakrelratetime(maxnnucl,maxrealizations,maxsubarea)

parameter (maxdisruptiveevents = 20)
dimension wpsfailedtime(4,maxdisruptiveevents,2)

```

12/14

```
dimension cumtempwp(maxntime)
dimension cumrelhumwp(maxntime)
dimension cumclwp(maxntime)

dimension percinfil( maxnsubarea, maxntime, 2)
dimension cumpercinfil( maxntime, 3)

dimension samplparvalue(maxpdf)

character*80 aline

cc rwr no dose      dimension dwdcffrac(43)
dimension dwdcf(43)

dimension aveqm3peryrz(maxnsubarea)

dimension amolepermtuejected(43)

cc rwr end of modifications

cc rwr 6/16/97 modified to remove forstudy message
dimension dummyarray(maxntime,1)

cc rwr 7/18/97 modifications for header
cc rwr 8/3/97 columns names added
cc      character columnheader
cc rwr 9/3/97 modified to remove for_study messages
cc      character columnheader1
cc      character columnheader2
cc      character*1600 columnheader1
cc      character*1600 columnheader2

cc rwr 9/3/97 modified to remove for_study messages
character*320 mes3

character*60 longname
character*8  shortname
character*60 parname
character*8  junk2
cc rwr 8/3/97 columns names added
cc      character*3 columnname
cc      character*2 columnname1
cc      character*5 columnname2
cc      character*5 columnname3
cc      character*4 columnname4

cc asl 6/9/97 modifications for output
character*60 formatstr
cc asl 6/9/97 End of output modifications

character*6 message

cc rwr 5/16/97 modified because included in setfiles.i
cc      character*80 title1, title2
cc      character*24 datetim
cc      character*24 ctime

external ctime
integer      idatetim, time

external time

integer sh
external sh
```

13/14

```
external ivaluesp
external valuesp
external ranl

external iaddconsmv
external isetconmv

cc rwr 6/7/97 modifications for output
    logical lexist
cc rwr end of modifications

cc rwj 12-12-97
    logical isfirstsubarea

cc NOTE: using SUN Fortran compiler routine to get DATE & TIME info
    idatetim = time()
    datetim = ctime(idatetim)

cc rwr 6/12/97 modifications for output
    iunitashplumeout = igetunitnumber('exec  ')
cc rwr end of modifications

    itpainp = igetunitnumber('exec  ')
    istop = igetunitnumber('exec  ')

    open( unit=itpainp, file='tpa.inp', status='old' )

    write(*,10001)
cc rwr 6/16/97 modified to remove forstudy message
cc    print (('10x,a)'), 'exec: Welcome to TPA Version 3.xxx'
    print (('10x,a)'), 'exec: Welcome to TPA Version 3.1.2'

cc rwr 1/14/98 modified screen print for time and date of run
    print (('8x,a,a)'), 'Job started: ',datetim

    write(*,10001)
10001 format(72('='))
10002 format(72('-'))

cc    print *, ' exec: calling reader '
    call reader( itpainp, title1, title2,
& maxntime, ntim, tim(1),
& maxnnucl, nnucl, names(1) )

cc rwr 8/28/97 added to close "tpa.inp" after reader
    close(itpainp)

cc rwr 6/7/97 modifications for output
    inquire( file='npkdoset.res', EXIST=lexist)
    if (lexist) istatus = sh( 'rm npkdoset.res' )
    iunitnpkdoset = igetunitnumber('exec  ')
    open( unit=iunitnpkdoset,file='npkdoset.res',status='unknown' )

cc rwr 7/18/97 modifications for header
    call clearchar( 160, mes1)
    call clearchar( 160, mes2)
    mes1 = ' Peak Dose and Corresponding Time by Nuclide'
    mes2 = ' Values for Each Vector '
    call writehead(iunitnpkdoset,title1,title2,datetim,mes1,mes2)
cc rwr 7/18/97 end of header modifications

    inquire( file='totdose.res', EXIST=lexist)
    if (lexist) istatus = sh( 'rm totdose.res' )
    iunittotdose = igetunitnumber('exec  ')
```

14/14

```
open( unit=iunittotdose,file='totdose.res',status='unknown')

cc rwr 7/18/97 modifications for header
call clearchar( 160, mes1)
call clearchar( 160, mes2)
mes1 = ' Total Dose for All Pathways, All Nuclides, and All Times'
mes2 = ' Including Dilution Volume - Values for Each Vector '
call writehead(iunittotdose,title1,title2,datetim,mes1,mes2)
```

pcl5

JOB 312

snllhs.rpt.nrc

no 45 study mode

PRINT outs of
For Study output

For: cscherer
Date: Thu Jan 15 11:35:19 CST 1998
Submit queue: IF 1 / Ethernet / UHSW
Submitted: Wed Jan 15 13:00:29 1998
Started: Wed Jan 15 13:00:29 1998

Warning #700 snllhs.f,94: No interface for routine IEEE_HANDLER()
Warning #700 snllhs.f,404: No interface for routine EXIT()
Warning #72 snllhs.f,429: common /WORKX/ was not used in function LHS()
Warning #72 snllhs.f,491: common /SAMP/ was not used in function BANNER()
Warning #72 snllhs.f,597: common /SAMP/ was not used in function CHKDIM()
Warning #72 snllhs.f,677: common /PARAM/ was not used in function CHKSTR()
Warning #72 snllhs.f,677: common /SAMP/ was not used in function CHKSTR()
Warning #72 snllhs.f,849: common /SAMP/ was not used in function CHLSKY()
Warning #543 snllhs.f,905: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #72 snllhs.f,947: common /SAMP/ was not used in function CMCRD()
Warning #72 snllhs.f,996: common /SAMP/ was not used in function CONSTA()
Warning #72 snllhs.f,1598: common /SAMP/ was not used in function HISTO()
Warning #72 snllhs.f,1664: common /SAMP/ was not used in function HPSRT()
Error #171 snllhs.f,2165: Local name OUTPUT was referenced but never set
Warning #72 snllhs.f,2451: common /SAMP/ was not used in function MATINV()
Warning #543 snllhs.f,2518: loss of precision in assignment: REAL*4 to INTEGER*4
Warning #626 snllhs.f,2567: differing number of MATINV() arguments; intrf. vs. call
Warning #72 snllhs.f,2746: common /SAMP/ was not used in function OUTCRD()
Warning #72 snllhs.f,3007: common /SAMP/ was not used in function PMTRX()
Warning #72 snllhs.f,3076: common /PARAM/ was not used in function POSDEF()
Warning #72 snllhs.f,3076: common /SAMP/ was not used in function POSDEF()
Warning #719 snllhs.f,3298: expecting array (to set) - string constant passed in arg #1 of XERROR()
Warning #684 snllhs.f,3298: mixed strings and arithmetic objects in arg #1 of XERROR()
Warning #713 snllhs.f,3298: expecting lvalue as scalar argument - constant/expr passed in arg #2 of XERROR()
Warning #713 snllhs.f,3298: expecting lvalue as scalar argument - constant/expr passed in arg #3 of XERROR()
Warning #713 snllhs.f,3298: expecting lvalue as scalar argument - constant/expr passed in arg #4 of XERROR()
Warning #72 snllhs.f,3393: common /SAMP/ was not used in function RANKER()
Warning #436 snllhs.f,3473: parameter PC char constant definition will be truncated
Error #171 snllhs.f,3677: Local name CDUM1 was referenced but never set
Error #171 snllhs.f,3677: Local name CDUM2 was referenced but never set
Warning #72 snllhs.f,3677: common /SAMP/ was not used in function RDPAR()
Warning #103 snllhs.f,3865: expecting REAL constant in DATA - not CHARACTER
Warning #72 snllhs.f,3889: common /SAMP/ was not used in function SETDEF()
Warning #719 snllhs.f,3999: expecting array (to set) - string constant passed in arg #1 of XERROR()
Warning #684 snllhs.f,3999: mixed strings and arithmetic objects in arg #1 of XERROR()
Warning #713 snllhs.f,3999: expecting lvalue as scalar argument - constant/expr passed in arg #2 of XERROR()
Warning #713 snllhs.f,3999: expecting lvalue as scalar argument - constant/expr passed in arg #3 of XERROR()
Warning #713 snllhs.f,3999: expecting lvalue as scalar argument - constant/expr passed in arg #4 of XERROR()
Warning #719 snllhs.f,4001: expecting array (to set) - string constant passed in arg #1 of XERROR()
Warning #684 snllhs.f,4001: mixed strings and arithmetic objects in arg #1 of XERROR()
Warning #713 snllhs.f,4001: expecting lvalue as scalar argument - constant/expr passed in arg #2 of XERROR()
Warning #713 snllhs.f,4001: expecting lvalue as scalar argument - constant/expr passed in arg #3 of XERROR()
Warning #713 snllhs.f,4001: expecting lvalue as scalar argument - constant/expr passed in arg #4 of XERROR()
Warning #72 snllhs.f,4690: common /SAMP/ was not used in function VIF()
Warning #72 snllhs.f,4740: common /SAMP/ was not used in function WRTCRD()
Warning #72 snllhs.f,4857: common /SAMP/ was not used in function WRTPAR()
Warning #684 snllhs.f,5138: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5139: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5141: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5143: mixed strings and arithmetic objects in arg #1 of XERSAV()
Warning #684 snllhs.f,5144: mixed strings and arithmetic objects in arg #1 of XERABT()
Warning #684 snllhs.f,5169: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5172: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5173: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5175: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5176: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5209: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5211: mixed strings and arithmetic objects in arg #1 of XERPRT()
Warning #684 snllhs.f,5213: mixed strings and arithmetic objects in arg #1 of XERSAV()
Warning #543 snllhs.f,5615: loss of precision in assignment: REAL*8 to REAL*4

Warning #543 snllhs.f,5617: loss of precision in assignment: REAL*8 to REAL*4
Warning #70 snllhs.f,5702: common /EVAL/ changed relative to prior use
Warning #71 snllhs.f,5702: common /EVAL/ changed relative to first usage
Warning #71 snllhs.f,5727: common /EVAL/ changed relative to first usage

pcl5

JOB 311

4 study mode

snllhs.rpt.mods

For Study
print outs

For: cscherer
Date: Thu Jan 15 11:35:16 CST 1998
Submit queue: IF 1 / Ethernet / UHSW
Submitted: Wed Jan 15 13:00:15 1998
Started: Wed Jan 15 13:00:15 1998

Warning #700 snllhs.f,99: No interface for routine IEEE_HANDLER()
Warning #700 snllhs.f,409: No interface for routine EXIT()
Warning #719 snllhs.f,3374: expecting array (to set) - string constant passed in arg #1 of XERROR()
Warning #713 snllhs.f,3374: expecting lvalue as scalar argument - constant/expr passed in arg #2 of XERROR()
Warning #713 snllhs.f,3374: expecting lvalue as scalar argument - constant/expr passed in arg #3 of XERROR()
Warning #713 snllhs.f,3374: expecting lvalue as scalar argument - constant/expr passed in arg #4 of XERROR()
Warning #719 snllhs.f,4093: expecting array (to set) - string constant passed in arg #1 of XERROR()
Warning #713 snllhs.f,4093: expecting lvalue as scalar argument - constant/expr passed in arg #2 of XERROR()
Warning #713 snllhs.f,4093: expecting lvalue as scalar argument - constant/expr passed in arg #3 of XERROR()
Warning #713 snllhs.f,4093: expecting lvalue as scalar argument - constant/expr passed in arg #4 of XERROR()
Warning #719 snllhs.f,4096: expecting array (to set) - string constant passed in arg #1 of XERROR()
Warning #713 snllhs.f,4096: expecting lvalue as scalar argument - constant/expr passed in arg #2 of XERROR()
Warning #713 snllhs.f,4096: expecting lvalue as scalar argument - constant/expr passed in arg #3 of XERROR()
Warning #713 snllhs.f,4096: expecting lvalue as scalar argument - constant/expr passed in arg #4 of XERROR()

pcl5

JOB 300

snllhs.dif

diff snllhs.f. nrc snllhs.f. mods
↑
contains last set of 4 study -
directed mods

no new code; diffs are declaring
double precision variables, changing
Hollerith strings to character strings
and removing common and variables
which aren't used.

For: cscherer
Date: Thu Jan 15 11:31:10 CST 1998

Submit queue: IF 1 / Ethernet / UHSW
Submitted: Wed Jan 15 12:56:19 1998
Started: Wed Jan 15 12:56:19 1998



QMS 3825 Print System

QMS 3825 Print System

```

62,64c62,66
<      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
<      1      ICORR,IDIST(NVAR),IRP
<      COMMON/SAMP/X(NMAX*NVAR)
---
>      CHARACTER TITLE(LENT)
>      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
>      1      ICORR,IDIST(NVAR),IRP, TITLE
>      REAL*8 X(NMAX*NVAR), XTMP(NMAX*NVAR)
>      COMMON/SAMP/X
70,73c72,79
<      COMMON/CMATR/CORR((NVAR*(NVAR+1))/2),LCM(NVAR),NCM
<      COMMON/RANK/XV(NMAX),RXV(NMAX),IWK(NMAX)
<      COMMON/WORKX/XX(NMAX*NVAR)
<      COMMON/WORKC/Q((NVAR*(NVAR+1))/2),S((NVAR*(NVAR+1))/2)
---
>      REAL*8 CORR((NVAR*(NVAR+1))/2)
>      COMMON/CMATR/CORR,LCM(NVAR),NCM
>      REAL*8 XV(NMAX)
>      COMMON/RANK/XV,RXV(NMAX),IWK(NMAX)
> C CSS 9/3/97 FORSTUDY: common WORKX not used
> C CSS COMMON/WORKX/XX(NMAX*NVAR)
>      REAL*8 S((NVAR*(NVAR+1))/2), Q((NVAR*(NVAR+1))/2)
>      COMMON/WORKC/Q, S
75d80
<      real xtmp(NVAR*NMAX)
82,83c87,88
<      CHARACTER*64 fpath
<      CHARACTER*64 lgdfil
---
> C CSS 9/3/97 not used      CHARACTER*64 fpath
> C CSS 9/3/97 not used      CHARACTER*64 lgdfil
90c95
<      INTEGER pos
---
> C CSS 9/3/97 not used      INTEGER pos
404c409
<      CALL EXIT
---
>      CALL EXIT(0)
454,456c459,463
<      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
<      1      ICORR,IDIST(NVAR),IRP
<      COMMON/SAMP/X(NMAX*NVAR)
---
>      CHARACTER TITLE(LENT)
>      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
>      1      ICORR,IDIST(NVAR),IRP, TITLE
> C CSS 9/3/97 FORSTUDY: common SAMP not used
> C CSS COMMON/SAMP/X(NMAX*NVAR)
542,544c549,553
<      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
<      1      ICORR,IDIST(NVAR),IRP
<      COMMON/SAMP/X(NMAX*NVAR)
---
>      CHARACTER TITLE(LENT)
>      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
>      1      ICORR,IDIST(NVAR),IRP, TITLE
> C CSS 9/3/97 FORSTUDY: common SAMP not used
> C CSS COMMON/SAMP/X(NMAX*NVAR)

```

613,615c622,625

```
< COMMON/PARAM/TITLE(LENT), ISEED, N, NV, IRS, ICM, NREP, IDATA, IHIST,
< 1 ICORR, IDIST(NVAR), IRP
< COMMON/SAMP/X(NMAX*NVAR)
---
```

```
> C CSS 9/3/97 FORSTUDY: commons PARAM & SAMP not used
> C CSS COMMON/PARAM/TITLE(LENT), ISEED, N, NV, IRS, ICM, NREP, IDATA, IHIST,
> C CSS1 ICORR, IDIST(NVAR), IRP
> C CSS COMMON/SAMP/X(NMAX*NVAR)
```

809,811c819,823

```
< COMMON/PARAM/TITLE(LENT), ISEED, N, NV, IRS, ICM, NREP, IDATA, IHIST,
< 1 ICORR, IDIST(NVAR), IRP
< COMMON/SAMP/X(NMAX*NVAR)
---
```

```
> CHARACTER TITLE(LENT)
> COMMON/PARAM/ISEED, N, NV, IRS, ICM, NREP, IDATA, IHIST,
> 1 ICORR, IDIST(NVAR), IRP, TITLE
> C CSS 9/3/97 FORSTUDY: common SAMP not used
> C CSS COMMON/SAMP/X(NMAX*NVAR)
```

822,823c834,837

```
< COMMON/CMATR/CORR((NVAR*(NVAR+1))/2), LCM(NVAR), NCM
< COMMON/WORKC/Q((NVAR*(NVAR+1))/2), S((NVAR*(NVAR+1))/2)
---
```

```
> REAL*8 CORR((NVAR*(NVAR+1))/2)
> COMMON/CMATR/CORR, LCM(NVAR), NCM
> REAL*8 S((NVAR*(NVAR+1))/2), Q((NVAR*(NVAR+1))/2)
> COMMON/WORKC/Q, S
```

865,867c879,883

```
< COMMON/PARAM/TITLE(LENT), ISEED, N, NV, IRS, ICM, NREP, IDATA, IHIST,
< 1 ICORR, IDIST(NVAR), IRP
< COMMON/SAMP/X(NMAX*NVAR)
---
```

```
> CHARACTER TITLE(LENT)
> COMMON/PARAM/ISEED, N, NV, IRS, ICM, NREP, IDATA, IHIST,
> 1 ICORR, IDIST(NVAR), IRP, TITLE
> C CSS 9/3/97 FORSTUDY: common SAMP not used
> C CSS COMMON/SAMP/X(NMAX*NVAR)
```

879c895,896

```
< COMMON/CMATR/CORR((NVAR*(NVAR+1))/2), LCM(NVAR), NCM
---
```

```
> REAL*8 CORR((NVAR*(NVAR+1))/2)
> COMMON/CMATR/CORR, LCM(NVAR), NCM
```

881c898,899

```
< DIMENSION RIJ(NCVAR*2), IJCVAR(2*NCVAR)
---
```

```
> DIMENSION IJCVAR(2*NCVAR)
> REAL*8 RIJ(NCVAR*2)
```

903a922

```
> C CSS 9/3/97 FORSTUDY: make implicit truncation explicit
```

905c924

```
< 120 IJCVAR(I)=RIJ(I)
---
```

```
> 120 IJCVAR(I)=int(RIJ(I))
```

963,965c982,986

```
< COMMON/PARAM/TITLE(LENT), ISEED, N, NV, IRS, ICM, NREP, IDATA, IHIST,
< 1 ICORR, IDIST(NVAR), IRP
< COMMON/SAMP/X(NMAX*NVAR)
---
```

```
> CHARACTER TITLE(LENT)
> COMMON/PARAM/ISEED, N, NV, IRS, ICM, NREP, IDATA, IHIST,
```

```

>      1      ICORR,IDIST(NVAR),IR,  TITLE
> C CSS 9/3/97 FORSTUDY:  common SAMP not used
> C CSS COMMON/SAMP/X(NMAX*NVAR)
1012,1014c1033,1037
<      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
<      1      ICORR,IDIST(NVAR),IRP
<      COMMON/SAMP/X(NMAX*NVAR)
---
>      CHARACTER TITLE(LENT)
>      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
>      1      ICORR,IDIST(NVAR),IRP,  TITLE
>      REAL*8 X(NMAX*NVAR)
>      COMMON/SAMP/X
1027,1028c1050,1052
<      COMMON/CMATR/CORR((NVAR*(NVAR+1))/2),LCM(NVAR),NCM
<      DIMENSION XM(NVAR),SSQ(NVAR)
---
>      REAL*8 CORR((NVAR*(NVAR+1))/2)
>      COMMON/CMATR/CORR,LCM(NVAR),NCM
>      REAL*8 XM(NVAR),SSQ(NVAR)
1096,1098c1120,1124
<      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
<      1      ICORR,IDIST(NVAR),IRP
<      COMMON/SAMP/X(NMAX*NVAR)
---
>      CHARACTER TITLE(LENT)
>      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
>      1      ICORR,IDIST(NVAR),IRP,  TITLE
>      REAL*8 X(NMAX*NVAR)
>      COMMON/SAMP/X
1110c1136,1137
<      COMMON/CMATR/CORR((NVAR*(NVAR+1))/2),LCM(NVAR),NCM
---
>      REAL*8 CORR((NVAR*(NVAR+1))/2)
>      COMMON/CMATR/CORR,LCM(NVAR),NCM
1112c1139,1140
<      COMMON/RANK/XV(NMAX),RXV(NMAX),IWK(NMAX)
---
>      REAL*8 XV(NMAX)
>      COMMON/RANK/XV,RXV(NMAX),IWK(NMAX)
1152,1154c1180,1184
<      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
<      1      ICORR,IDIST(NVAR),IRP
<      COMMON/SAMP/X(NMAX*NVAR)
---
>      CHARACTER TITLE(LENT)
>      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
>      1      ICORR,IDIST(NVAR),IRP,  TITLE
>      REAL*8 X(NMAX*NVAR)
>      COMMON/SAMP/X
1166c1196,1197
<      COMMON/RANK/XV(NMAX),RXV(NMAX),IWK(NMAX)
---
>      REAL*8 XV(NMAX)
>      COMMON/RANK/XV,RXV(NMAX),IWK(NMAX)
1261c1292,1293
<      COMMON/CMATR/CORR((NVAR*(NVAR+1))/2),LCM(NVAR),NCM
---
>      REAL*8 CORR((NVAR*(NVAR+1))/2),  TOL,  DSUM,  DPIV
>      COMMON/CMATR/CORR,LCM(NVAR),NCM

```

10/52

```
1315c1347,1348
< COMMON/CMATR/CORR((NVAR*(NVAR+1))/2),LCM(NVAR),NCM
---
> REAL*8 CORR((NVAR*(NVAR+1))/2), DIN, WORK
> COMMON/CMATR/CORR,LCM(NVAR),NCM
1407,1409c1440,1444
< COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
< 1 ICORR,IDIST(NVAR),IRP
< COMMON/SAMP/X(NMAX*NVAR)
---
> CHARACTER TITLE(LENT)
> COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
> 1 ICORR,IDIST(NVAR),IRP, TITLE
> REAL*8 X(NMAX*NVAR)
> COMMON/SAMP/X
1421,1422c1456,1459
< COMMON/CMATR/CORR((NVAR*(NVAR+1))/2),LCM(NVAR),NCM
< COMMON/PDMAT/Z(NVAR,NVAR),D(NVAR)
---
> REAL*8 CORR((NVAR*(NVAR+1))/2)
> COMMON/CMATR/CORR,LCM(NVAR),NCM
> REAL*8 D(NVAR), Z(NVAR,NVAR)
> COMMON/PDMAT/Z,D
1492,1494c1529,1533
< COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
< 1 ICORR,IDIST(NVAR),IRP
< COMMON/SAMP/X(NMAX*NVAR)
---
> CHARACTER TITLE(LENT)
> COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
> 1 ICORR,IDIST(NVAR),IRP, TITLE
> C CSS 9/3/97 FORSTUDY: common SAMP not used
> C CSS COMMON/SAMP/X(NMAX*NVAR)
1507c1546,1548
< COMMON/RANK/XV(NMAX),RXV(NMAX),IWK(NMAX)
---
> REAL*8 CELL, POINT, SIZE, XMEAN, XVAR, POWER
> REAL*8 XV(NMAX), SUM, SUMSQ, FMED, R, TEMP
> COMMON/RANK/XV,RXV(NMAX),IWK(NMAX)
1614,1616c1655,1659
< COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
< 1 ICORR,IDIST(NVAR),IRP
< COMMON/SAMP/X(NMAX*NVAR)
---
> CHARACTER TITLE(LENT)
> COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
> 1 ICORR,IDIST(NVAR),IRP, TITLE
> C CSS 9/3/97 FORSTUDY: common SAMP not used
> C CSS COMMON/SAMP/X(NMAX*NVAR)
1628c1671,1672
< COMMON/RANK/XV(NMAX),RXV(NMAX),IWK(NMAX)
---
> REAL*8 XV(NMAX), XHOLD
> COMMON/RANK/XV,RXV(NMAX),IWK(NMAX)
1680,1682c1724,1728
< COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
< 1 ICORR,IDIST(NVAR),IRP
< COMMON/SAMP/X(NMAX*NVAR)
---
> CHARACTER TITLE(LENT)
```


11/52

```
> COMMON/PARAM/ISEED,N,NV,IRS,ICM,N1 IDATA,IHIST,
> 1 ICORR,IDIST(NVAR),IRP, TITLE
> REAL*8 X(NMAX*NVAR)
> COMMON/SAMP/X
1700c1746,1747
< COMMON/RANK/XV(NMAX),RXV(NMAX),IWK(NMAX)
---
> REAL*8 XV(NMAX)
> COMMON/RANK/XV,RXV(NMAX),IWK(NMAX)
2154c2201,2202
< WRITE(OUTPUT,9000)
---
> C CSS 9/3/97 FORSTUDY: OUTPUT is never set; comment out write
> C CSS WRITE(OUTPUT,9000)
2236,2238c2284,2286
< REAL D(N),E(N),Z(NM,N)
< REAL B,C,F,G,P,R,S,S1,S2
< REAL PYTHAG
---
> REAL*8 D(N),E(N),Z(NM,N)
> REAL*8 B,C,F,G,P,R,S,S1,S2
> REAL*8 PYTHAG
2265c2313
< R = PYTHAG(G,1.0E0)
---
> R = PYTHAG(G,1.0D0)
2417,2419c2465,2469
< COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
< 1 ICORR,IDIST(NVAR),IRP
< COMMON/SAMP/X(NMAX*NVAR)
---
> CHARACTER TITLE(LENT)
> COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
> 1 ICORR,IDIST(NVAR),IRP, TITLE
> C CSS 9/3/97 FORSTUDY: common SAMP not used
> C CSS COMMON/SAMP/X(NMAX*NVAR)
2430c2480,2481
< COMMON/WORKC/Q((NVAR*(NVAR+1))/2),S((NVAR*(NVAR+1))/2)
---
> REAL*8 S((NVAR*(NVAR+1))/2), Q((NVAR*(NVAR+1))/2), TEMP
> COMMON/WORKC/Q, S
2467,2469c2518,2522
< COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
< 1 ICORR,IDIST(NVAR),IRP
< COMMON/SAMP/X(NMAX*NVAR)
---
> CHARACTER TITLE(LENT)
> COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
> 1 ICORR,IDIST(NVAR),IRP, TITLE
> REAL*8 X(NMAX*NVAR)
> COMMON/SAMP/X
2481c2534,2535
< COMMON/CMATR/CORR((NVAR*(NVAR+1))/2),LCM(NVAR),NCM
---
> REAL*8 CORR((NVAR*(NVAR+1))/2), ELEM, CMX, CMXOLD
> COMMON/CMATR/CORR,LCM(NVAR),NCM
2483c2537,2538
< COMMON/RANK/XV(NMAX),RXV(NMAX),IWK(NMAX)
---
> REAL*8 XV(NMAX)
```

```

> COMMON/RANK/XV,RXV(NMAX),IWK(NMAX,
2485c2540,2541
< COMMON/WORKC/Q((NVAR*(NVAR+1))/2),S((NVAR*(NVAR+1))/2)
---
> REAL*8 S((NVAR*(NVAR+1))/2), Q((NVAR*(NVAR+1))/2)
> COMMON/WORKC/Q, S
2518c2574,2575
< JJ=IR*RAN(ISEED)+1
---
> C CSS 9/3/97 FORSTUDY: make implicit truncation explicit
> JJ=int(IR*RAN(ISEED)+1)
2567c2624,2628
< CALL MATINV(NV)
---
> C CSS 9/3/97 FORSTUDY: MATINV doesn't accept any arguments; NV is contained
> C CSS in common PARAM which is contained in this function and MATINV
> C CSS CALL MATINV(NV)
> CALL MATINV
>
2639,2641c2700,2704
< COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
< 1 ICORR,IDIST(NVAR),IRP
< COMMON/SAMP/X(NMAX*NVAR)
---
> CHARACTER TITLE(LENT)
> COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
> 1 ICORR,IDIST(NVAR),IRP, TITLE
> REAL*8 X(NMAX*NVAR)
> COMMON/SAMP/X
2699,2701c2762,2766
< COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
< 1 ICORR,IDIST(NVAR),IRP
< COMMON/SAMP/X(NMAX*NVAR)
---
> CHARACTER TITLE(LENT)
> COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
> 1 ICORR,IDIST(NVAR),IRP, TITLE
> C CSS 9/3/97 FORSTUDY: common SAMP not used
> C CSS COMMON/SAMP/X(NMAX*NVAR)
2762,2764c2827,2831
< COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
< 1 ICORR,IDIST(NVAR),IRP
< COMMON/SAMP/X(NMAX*NVAR)
---
> CHARACTER TITLE(LENT)
> COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
> 1 ICORR,IDIST(NVAR),IRP, TITLE
> REAL*8 X(NMAX*NVAR)
> COMMON/SAMP/X
2918,2920c2985,2989
< COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
< 1 ICORR,IDIST(NVAR),IRP
< COMMON/SAMP/X(NMAX*NVAR)
---
> CHARACTER TITLE(LENT)
> COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
> 1 ICORR,IDIST(NVAR),IRP, TITLE
> C CSS 9/3/97 FORSTUDY: common SAMP not used
> C CSS COMMON/SAMP/X(NMAX*NVAR)
2933c3002,3003

```

```

<      COMMON/CMATR/CORR((NVAR*(NVAR+1)))      LCM(NVAR),NCM
---
>      REAL*8 CORR((NVAR*(NVAR+1))/2)
>      COMMON/CMATR/CORR,LCM(NVAR),NCM
3023,3025c3093,3097
<      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
<      1      ICORR,IDIST(NVAR),IRP
<      COMMON/SAMP/X(NMAX*NVAR)
---
> C CSS 9/3/97 FORSTUDY:  common PARAM not used
> C CSS COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
> C CSS1      ICORR,IDIST(NVAR),IRP
> C CSS 9/3/97 FORSTUDY:  common SAMP not used
> C CSS COMMON/SAMP/X(NMAX*NVAR)
3033,3035c3105,3109
<      COMMON/CMATR/CORR((NVAR*(NVAR+1))/2),LCM(NVAR),NCM
<      COMMON/PDMAT/Z(NVAR,NVAR),D(NVAR)
<      COMMON/WORK/WK((NVAR*(NVAR+1))/2)
---
>      REAL*8 CORR((NVAR*(NVAR+1))/2)
>      COMMON/CMATR/CORR,LCM(NVAR),NCM
>      REAL*8 D(NVAR), Z(NVAR,NVAR), WK((NVAR*(NVAR+1))/2)
>      COMMON/PDMAT/Z,D
>      COMMON/WORK/WK
3077c3151
<      REAL FUNCTION PYTHAG(A,B)
---
>      REAL*8 FUNCTION PYTHAG(A,B)
3083c3157
<      REAL A,B
---
>      REAL*8 A,B
3085c3159
<      REAL P,Q,R,S,T
---
>      REAL*8 P,Q, R,S,T
3298c3372,3374
<      1      CALL XERROR (25HR1MACH -- I OUT OF BOUNDS,25,1,2)
---
> C CSS 9/3/97 FORSTUDY:  replace Hollerith string w/ character string
> C CSS1      CALL XERROR (25HR1MACH -- I OUT OF BOUNDS,25,1,2)
>      1      CALL XERROR ('R1MACH -- I OUT OF BOUNDS',25,1,2)
3322,3324c3398,3402
<      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
<      1      ICORR,IDIST(NVAR),IRP
<      COMMON/SAMP/X(NMAX*NVAR)
---
>      CHARACTER TITLE(LENT)
>      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
>      1      ICORR,IDIST(NVAR),IRP, TITLE
> C CSS 9/3/97 FORSTUDY:  common SAMP not used
> C CSS COMMON/SAMP/X(NMAX*NVAR)
3336c3414,3415
<      COMMON/RANK/XV(NMAX),RXV(NMAX),IWK(NMAX)
---
>      REAL*8 XV(NMAX), XHOLD
>      COMMON/RANK/XV,RXV(NMAX),IWK(NMAX)
3409,3411c3488,3492
<      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
<      1      ICORR,IDIST(NVAR),IRP

```

14/52

```
<      COMMON/SAMP/X(NMAX*NVAR)
---
>      CHARACTER TITLE(LENT)
>      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
>      1          ICORR,IDIST(NVAR),IRP, TITLE
> C CSS 9/3/97 FORSTUDY:  common SAMP not used
> C CSS COMMON/SAMP/X(NMAX*NVAR)
3429a3511
> C CSS 9/3/97 FORSTUDY:  PC needs to be at least 4 characters long
3431c3513
<      1          PC,PCMINP*18,LCMINP*29,POUT*7,PBETA*5,PNOR*7,PLNOR*10,
---
>      1          PC*4,PCMINP*18,LCMINP*29,POUT*7,PBETA*5,PNOR*7,PLNOR*10,
3478a3561,3564
> C CSS 9/3/97 FORSTUDY:  CDUM1 & CDUM2 never set; printed out in chkdim;
> C CSS                      default to blank
>      CDUM1 = ' '
>      CDUM2 = ' '
3851,3853c3937,3941
<      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
<      1          ICORR,IDIST(NVAR),IRP
<      COMMON/SAMP/X(NMAX*NVAR)
---
>      CHARACTER TITLE(LENT)
>      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
>      1          ICORR,IDIST(NVAR),IRP, TITLE
> C CSS 9/3/97 FORSTUDY:  common SAMP not used
> C CSS COMMON/SAMP/X(NMAX*NVAR)
3864a3953,3954
> C CSS 9/3/97 FORSTUDY:  delcare blank as character variable
>      CHARACTER*1 blank
3870c3960,3961
<      10 TITLE(I)=blank
---
> C CSS 9/3/97 FORSTUDY:  use conversion function  ??
>      10 TITLE(I)=(blank)
3891c3982
<      DIMENSION XV(N)
---
>      REAL*8 XV(N), A
3995,3996c4086,4088
<      INTEGER I,INFO,J,K,LDV,M,N
<      REAL A((N*(N+1))/2),E(N),V(LDV,N),WORK(1)
---
>      INTEGER I,INFO,J,LDV,M,N, NV
>      REAL*8 A((N*(N+1))/2), E(N)
>      REAL*8 V(LDV,N),WORK(1)
3999c4091,4093
<      IF(N .GT. LDV) CALL XERROR(17HSSPEV-N .GT. LDV.,17,1,1)
---
> C CSS 9/3/97 FORSTUDY:  replace Hollerith string w/ character string
> C CSS  IF(N .GT. LDV) CALL XERROR(17HSSPEV-N .GT. LDV.,17,1,1)
>      IF(N .GT. LDV) CALL XERROR('SSPEV-N .GT. LDV.',17,1,1)
4001c4095,4096
<      IF(N .LT. 1) CALL XERROR(14HSSPEV-N .LT. 1,14,2,1)
---
> C CSS 9/3/97 FORSTUDY:  replace Hollerith string w/ character string
>      IF(N .LT. 1) CALL XERROR('SSPEV-N .LT. 1',14,2,1)
4087,4089c4182,4184
<      REAL D(N),E2(N)
```

```

<      REAL B,C,F,G,H,P,R,S,MACHEP
<      REAL PYTHAG
---
>      REAL*8 D(N),E2(N)
>      REAL*8 B,C,F,G,H,P,R,S,MACHEP
>      REAL*8 PYTHAG
4129c4224
<      R = PYTHAG(P,1.0E0)
---
>      R = PYTHAG(P,1.0D0)
4244,4245c4339,4341
<      REAL A(NV),Z(NM,M)
<      REAL H,S
---
>      REAL*8 A(NV)
>      REAL*8 Z(NM,M)
>      REAL*8 H,S
4339,4340c4435,4437
<      REAL A(NV),D(N),E(N),E2(N)
<      REAL F,G,H,HH,SCALE
---
>      REAL*8 A(NV)
>      REAL*8 D(N),E(N),E2(N)
>      REAL*8 F,G,H,HH,SCALE
4429,4431c4526,4530
<      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
<      1          ICORR,IDIST(NVAR),IRP
<      COMMON/SAMP/X(NMAX*NVAR)
---
>      CHARACTER TITLE(LENT)
>      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
>      1          ICORR,IDIST(NVAR),IRP, TITLE
>      REAL*8 X(NMAX*NVAR)
>      COMMON/SAMP/X
4473,4475c4572,4576
<      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
<      1          ICORR,IDIST(NVAR),IRP
<      COMMON/SAMP/X(NMAX*NVAR)
---
>      CHARACTER TITLE(LENT)
>      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
>      1          ICORR,IDIST(NVAR),IRP, TITLE
>      REAL*8 X(NMAX*NVAR)
>      COMMON/SAMP/X
4518,4520c4619,4623
<      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
<      1          ICORR,IDIST(NVAR),IRP
<      COMMON/SAMP/X(NMAX*NVAR)
---
>      CHARACTER TITLE(LENT)
>      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
>      1          ICORR,IDIST(NVAR),IRP, TITLE
>      REAL*8 X(NMAX*NVAR)
>      COMMON/SAMP/X
4532d4634
< C      COMMON/SAMP/X(NVAR*NMAX)
4557,4559c4659,4663
<      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
<      1          ICORR,IDIST(NVAR),IRP
<      COMMON/SAMP/X(NMAX*NVAR)

```

```

---
> CHARACTER TITLE(LENT)
> COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
> 1 ICORR,IDIST(NVAR),IRP, TITLE
> REAL*8 X(NMAX*NVAR)
> COMMON/SAMP/X
4593,4595c4697,4701
< COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
< 1 ICORR,IDIST(NVAR),IRP
< COMMON/SAMP/X(NMAX*NVAR)
---
> CHARACTER TITLE(LENT)
> COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
> 1 ICORR,IDIST(NVAR),IRP, TITLE
> REAL*8 X(NMAX*NVAR)
> COMMON/SAMP/X
4661,4663c4767,4771
< COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
< 1 ICORR,IDIST(NVAR),IRP
< COMMON/SAMP/X(NMAX*NVAR)
---
> CHARACTER TITLE(LENT)
> COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
> 1 ICORR,IDIST(NVAR),IRP, TITLE
> C CSS 9/3/97 FORSTUDY: common SAMP not used
> C CSS COMMON/SAMP/X(NMAX*NVAR)
4674c4782,4783
< COMMON/CMATR/CORR((NVAR*(NVAR+1))/2),LCM(NVAR),NCM
---
> REAL*8 CORR((NVAR*(NVAR+1))/2), CRK, CRKMX
> COMMON/CMATR/CORR,LCM(NVAR),NCM
4706,4709c4815,4819
< COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
< 1 ICORR,IDIST(NVAR),IRP
< COMMON/SAMP/X(NMAX*NVAR)
< C
---
> CHARACTER TITLE(LENT)
> COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
> 1 ICORR,IDIST(NVAR),IRP, TITLE
> C CSS 9/3/97 FORSTUDY: common SAMP not used
> C CSS COMMON/SAMP/X(NMAX*NVAR)
4756,4758c4866,4870
< COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
< 1 ICORR,IDIST(NVAR),IRP
< COMMON/SAMP/X(NMAX*NVAR)
---
> CHARACTER TITLE(LENT)
> COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
> 1 ICORR,IDIST(NVAR),IRP, TITLE
> C CSS 9/3/97 FORSTUDY: common SAMP not used
> C CSS COMMON/SAMP/X(NMAX*NVAR)
4858c4970,4972
< SUBROUTINE XERABT(MESSG,NMESSG)
---
> C CSS 9/3/97 FORSTUDY: accept character string instead of integer array
> C CSS SUBROUTINE XERABT(MESSG,NMESSG)
> SUBROUTINE XERABT(CMESSG,NMESSG)
4886c5000,5011
< DIMENSION MESSG(NMESSG)

```

```

---
> C CSS 9/3/97 FORSTUDY: declare char variable to accept character string
> C CSS      & equivalence it to integer array for Hollerith formatting;
> C CSS      set maximum length for message
>      PARAMETER (maxmsglen = 80)
>      DIMENSION MESSG (maxmsglen)
>      CHARACTER CMESSG(NMESSG), CBUFF(maxmsglen)
>      EQUIVALENCE (MESSG, CBUFF)
>      INTEGER icounter
> C CSS DIMENSION MESSG(NMESSG)
>      DO icounter=1,NMESSG
>          CBUFF(icounter) = CMESSG(icounter)
>      ENDDO
4946c5071,5073
<      SUBROUTINE XERPRT(MESSG,NMESSG)
---
> C CSS 9/3/97 FORSTUDY: accept character string instead of integer array
> C CSS SUBROUTINE XERPRT(MESSG,NMESSG)
>      SUBROUTINE XERPRT(CMESSG,NMESSG)
4965c5092,5101
<      DIMENSION MESSG(NMESSG)
---
> C CSS 9/3/97 FORSTUDY: declare char variable to accept character
> C CSS      string & equivalence it to integer array for Hollerith
> C CSS      formatting; set max length for message
>      PARAMETER (maxmsglen = 80)
>      DIMENSION MESSG(maxmsglen)
>      CHARACTER CMESSG(NMESSG), CBUFF(maxmsglen)
>      EQUIVALENCE (MESSG, CBUFF)
>      INTEGER icounter
> C CSS
> C CSS DIMENSION MESSG(NMESSG)
4974a5111,5115
> C CSS 9/3/97 FORSTUDY: set cbuff = to cmessg
>      DO icounter=1,NMESSG
>          CBUFF(icounter) = CMESSG(icounter)
>      ENDDO
> C CSS
5021c5162,5164
<      SUBROUTINE XERROR(MESSG,NMESSG,NERR,LEVEL)
---
> C CSS 9/3/97 FORSTUDY: accept character string instead of integer array
> C CSS SUBROUTINE XERROR(MESSG,NMESSG,NERR,LEVEL)
>      SUBROUTINE XERROR(CMESSG,NMESSG,NERR,LEVEL)
5068c5211,5213
<      DIMENSION MESSG(NMESSG)
---
> C CSS 9/3/97 FORSTUDY: character string, not integer array
> C CSS DIMENSION MESSG(NMESSG)
>      CHARACTER CMESSG(NMESSG)
5070c5215
<      CALL XERRWV(MESSG,NMESSG,NERR,LEVEL,0,0,0,0,0.,0.)
---
>      CALL XERRWV(CMESSG,NMESSG,NERR,LEVEL,0,0,0,0,0.,0.)
5077c5222,5224
<      SUBROUTINE XERRWV(MESSG,NMESSG,NERR,LEVEL,NI,I1,I2,NR,R1,R2)
---
> C CSS 9/3/97 FORSTUDY: accept character string instead of Hollerith string
> C CSS SUBROUTINE XERRWV(MESSG,NMESSG,NERR,LEVEL,NI,I1,I2,NR,R1,R2)
>      SUBROUTINE XERRWV(CMESSG,NMESSG,NERR,LEVEL,NI,I1,I2,NR,R1,R2)

```

5131c5278,5290

```

<      DIMENSION MESSG(NMESSG),LUN(5)
---
> C CSS 9/3/97 FORSTUDY:  declare messg as integer; set max message length
> C CSS DIMENSION MESSG(NMESSG),LUN(5)
>      PARAMETER (maxmsglen = 80)
>      DIMENSION LUN(5), MESSG(maxmsglen)
> C CSS DIMENSION MESSG(NMESSG),LUN(5)
> C CSS declare character string
>      CHARACTER CMESSG(NMESSG), CBUFF(maxmsglen)
>      EQUIVALENCE (MESSG, CBUFF)
>      INTEGER icounter
>      DO icounter = 1,NMESSG
>          CBUFF(icounter) = CMESSG(icounter)
>      ENDDO
> C CSS

```

5138,5139c5297,5301

```

<      IF (LKNTL.GT.0) CALL XERPRT(17HFATAL ERROR IN...,17)
<      CALL XERPRT(23HXERROR -- INVALID INPUT,23)
---
> C CSS 9/3/97 FORSTUDY:  replace Hollerith string w/ character string
> C CSS      IF (LKNTL.GT.0) CALL XERPRT(17HFATAL ERROR IN...,17)
>      IF (LKNTL.GT.0) CALL XERPRT('FATAL ERROR IN...',17)
> C CSS      CALL XERPRT(23HXERROR -- INVALID INPUT,23)
>      CALL XERPRT('XERROR -- INVALID INPUT',23)

```

5141c5303,5304

```

<      IF (LKNTL.GT.0) CALL XERPRT(29HJOB ABORT DUE TO FATAL ERROR.,
---
> C CSS      IF (LKNTL.GT.0) CALL XERPRT(29HJOB ABORT DUE TO FATAL ERROR.,
>      IF (LKNTL.GT.0) CALL XERPRT('JOB ABORT DUE TO FATAL ERROR.',

```

5143,5144c5306,5310

```

<      IF (LKNTL.GT.0) CALL XERSAV(1H ,0,0,0,KDUMMY)
<      CALL XERABT(23HXERROR -- INVALID INPUT,23)
---
> C CSS      IF (LKNTL.GT.0) CALL XERSAV(1H ,0,0,0,KDUMMY)
>      IF (LKNTL.GT.0) CALL XERSAV(' ',0,0,0,KDUMMY)
> C CSS 9/3/97 FORSTUDY:  replace Hollerith string w/ character string
> C CSS      CALL XERABT(23HXERROR -- INVALID INPUT,23)
>      CALL XERABT('XERROR -- INVALID INPUT',23)

```

5149c5315,5317

```

<      CALL XERSAV(MESSG,NMESSG,NERR,LEVEL,KOUNT)
---
> C CSS 9/3/97 FORSTUDY:  replace Hollerith string w/ character string
> C CSS CALL XERSAV(MESSG,NMESSG,NERR,LEVEL,KOUNT)
>      CALL XERSAV(CBUFF(1),NMESSG,NERR,LEVEL,KOUNT)

```

5172,5173c5340,5343

```

<      1(57HWARNING MESSAGE...THIS MESSAGE WILL ONLY BE PRINTED ONCE.,57)
<      IF (LLEVEL.EQ.0) CALL XERPRT(13HWARNING IN...,13)
---
>      1('WARNING MESSAGE...THIS MESSAGE WILL ONLY BE PRINTED ONCE.',57)
> C CSS1(57HWARNING MESSAGE...THIS MESSAGE WILL ONLY BE PRINTED ONCE.,57)
>      IF (LLEVEL.EQ.0) CALL XERPRT('WARNING IN...',13)
> C CSS      IF (LLEVEL.EQ.0) CALL XERPRT(13HWARNING IN...,13)

```

5175,5176c5345,5349

```

<      1      (23HRECOVERABLE ERROR IN...,23)
<      IF (LLEVEL.EQ.2) CALL XERPRT(17HFATAL ERROR IN...,17)
---
>      1      ('RECOVERABLE ERROR IN...',23)
> C CSS      1      (23HRECOVERABLE ERROR IN...,23)
> C CSS 9/3/97 FORSTUDY:  replace Hollerith string w/ character string

```


19/50

```

> C CSS      IF (LLEVEL.EQ.2) CALL XERPR  ,FATAL ERROR IN...,17)
>      IF (LLEVEL.EQ.2) CALL XERPRT('FATAL ERROR IN...',17)
5179c5352,5354
<      CALL XERPRT(MESSG,LMESSG)
---
> C CSS 9/3/97 FORSTUDY:  pass character string instead of Hollerith string
> C CSS      CALL XERPRT(MESSG,LMESSG)
>      CALL XERPRT(CBUFF,LMESSG)
5207a5383
> C CSS 9/3/97 FORSTUDY:  replace Hollerith string w/ character string
5209c5385,5386
<      1  (35HJOB ABORT DUE TO UNRECOVERED ERROR.,35)
---
> C CSS      1  (35HJOB ABORT DUE TO UNRECOVERED ERROR.,35)
>      1  ('JOB ABORT DUE TO UNRECOVERED ERROR.',35)
5211c5388,5389
<      1  (29HJOB ABORT DUE TO FATAL ERROR.,29)
---
> C CSS      1  (29HJOB ABORT DUE TO FATAL ERROR.,29)
>      1  ('JOB ABORT DUE TO FATAL ERROR.',29)
5213c5391,5393
<      CALL XERSAV(1H , -1,0,0,KDUMMY)
---
> C CSS 9/3/97 FORSTUDY:  replace Hollerith string w/ character string
> C CSS      CALL XERSAV(1H , -1,0,0,KDUMMY)
>      CALL XERSAV(' ' , -1,0,0,KDUMMY)
5217c5397,5399
<      CALL XERABT(MESSG,LMESSG)
---
> C CSS 9/3/97 FORSTUDY:  send char string instead of Hollerith string
> C CSS      CALL XERABT(MESSG,LMESSG)
>      CALL XERABT(CBUFF,LMESSG)
5224c5406,5408
<      SUBROUTINE XERSAV(MESSG,NMESSG,NERR,LEVEL,ICOUNT)
---
> C CSS 9/3/97 FORSTUDY:  accept character string instead of Hollerith
> C CSS      SUBROUTINE XERSAV(MESSG,NMESSG,NERR,LEVEL,ICOUNT)
>      SUBROUTINE XERSAV(CMESSG,NMESSG,NERR,LEVEL,ICOUNT)
5256a5441,5442
> C CSS 9/3/97 FORSTUDY:  declare char variable to accept char string
> C CSS      & equivalence it to integer array for Hollerith formatting
5257a5444,5446
>      CHARACTER*1 CMESSG, CBUFF
>      EQUIVALENCE (MESSG, CBUFF)
> C CSS
5276a5466,5468
> C CSS 9/3/97 FORSTUDY:  set cbuff = to cmessg
>      CBUFF = CMESSG
> C CSS
5410,5412c5602,5606
<      COMMON/PARAM/TITLE(LENT), ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
<      1      ICORR,IDIST(NVAR),IRP
<      COMMON/SAMP/X(NMAX*NVAR)
---
>      CHARACTER TITLE(LENT)
>      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
>      1      ICORR,IDIST(NVAR),IRP, TITLE
>      REAL*8 X(NMAX*NVAR)
>      COMMON/SAMP/X
5431c5625

```

```

<      DIMENSION XVAL(4),FREQ(4),CDF(5)
---
> C CSS 9/3/97 not used      DIMENSION XVAL(4),FREQ(4),CDF(5)
5564c5758,5759
<      REAL*8 D1,D2,D3,D4,D5,D6,VALUE
---
>      CHARACTER TITLE(LENT)
>      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
5566,5567d5760
<      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
<
5570c5763
<      1          ICORR,      IDIST(NVAR),IRP
---
>      1          ICORR,      IDIST(NVAR),IRP, TITLE
5572,5573c5765,5769
<      COMMON/SAMP/X(NVAR*NMAX)
<      COMMON/SIMU/SAMPLE(10*NSIM)
---
>      REAL*8 P,Q,T,R,S1,S2,VALUE, SAMPLE(10*NSIM), X(NVAR*NMAX)
>
>      COMMON/SAMP/X
> C CSS 9/3/97 FORSTUDY      COMMON/SIMU/SAMPLE(10*NSIM)
>      COMMON/SIMU/SAMPLE
5575c5771
<      COMMON/EVAL/D1,D2,D3,D4,D5,D6,VALUE
---
>      COMMON/EVAL/P,Q,T,R,S1,S2,VALUE
5733c5929,5930
<      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
---
>      CHARACTER TITLE(LENT)
>      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
5736c5933
<      1          ICORR,      IDIST(NVAR),IRP
---
>      1          ICORR,      IDIST(NVAR),IRP, TITLE

```

21/52

pcl5

JOB 309

tccdf.rpt

1/15/98 - no forstudy problems Carl Scherer

For: cscherer
Date: Thu Jan 15 11:33:46 CST 1998
Submit queue: IF 1 / Ethernet / UHSW
Submitted: Wed Jan 15 12:58:55 1998
Started: Wed Jan 15 12:58:55 1998

Warning #700 tccdf.f,339: No interface for routine FDATE()
Warning #700 tccdf.f,760: No interface for routine SORTQR()

} can be resolved by 22
inserting interface before.
into tccdf.nff

pcl5

JOB 307

releaset.rpt

1/15/98 no forstudy problems *Carl Scher*

For: cscherer
Date: Thu Jan 15 11:33:18 CST 1998

Submit queue: IF 1 / Ethernet / UHSW
Submitted: Wed Jan 15 12:58:27 1998
Started: Wed Jan 15 12:58:27 1998

Warning #700 releaset.f,363: No interface

routine FDATE()

24
can be resolved by inserting FDATE()
interface defn into releaset/ntf

pcl5

JOB 301

ashplume.rpt

no forstudy message on ashplume.f
1/15/98 Carl Scherer

For: cscherer
Date: Thu Jan 15 11:31:23 CST 1998

Submit queue: IF 1 / Ethernet / UHSW
Submitted: Wed Jan 15 12:57:06 1998
Started: Wed Jan 15 12:57:06 1998

pcl5

JOB 303

failt.rpt

1/15/98 - no forestry problems

Card S. Scherer

For: cscherer
Date: Thu Jan 15 11:32:24 CST 1998
Submit queue: IF 1 / Ethernet / UHSW
Submitted: Wed Jan 15 12:57:33 1998
Started: Wed Jan 15 12:57:33 1998



QMS 3825 Print System

QMS 3825 Print System

Warning #700 failt.f,175: No interface fo utine FDATE()

Warning #686 failt.f,877: different type and size used in arg #12 of RKQC()

/ can be resolved by inserting
inter^{face} defn. into .27
failt . ntf file



- this arg is a function name
which is declared a real*8

pcl5

JOB 305

nefmks.rpt . dev

4study mod

1/15/98

For: cscherer
Date: Thu Jan 15 11:32:51 CST 1998
Submit queue: IF 1 / Ethernet / UHSW
Submitted: Wed Jan 15 12:58:00 1998
Started: Wed Jan 15 12:58:00 1998



QMS 3825 Print System

QMS 3825 Print System

Warning #700 nefmks.f,474: No interface f outline FDATE()
Warning #103 nefmks.f,1118: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,1118: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,1119: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,1119: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,1119: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,1119: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,1119: expecting REAL constant in DATA - not REAL*8
Error #670 nefmks.f,2368: intrinsic MAX() args differ - REAL*4 vs. *8
Error #670 nefmks.f,2499: intrinsic MAX() args differ - REAL*4 vs. *8
Error #670 nefmks.f,2589: intrinsic MAX() args differ - REAL*4 vs. *8
Error #670 nefmks.f,2609: intrinsic MAX() args differ - REAL*4 vs. *8
Warning #103 nefmks.f,2720: expecting REAL constant in DATA - not REAL*8
Error #670 nefmks.f,2756: intrinsic MAX() args differ - REAL*4 vs. *8
Warning #543 nefmks.f,2756: loss of precision in assignment: REAL*8 to REAL*4
Error #670 nefmks.f,2869: intrinsic MAX() args differ - REAL*4 vs. *8
Warning #543 nefmks.f,2869: loss of precision in assignment: REAL*8 to REAL*4
Error #670 nefmks.f,2966: intrinsic MAX() args differ - REAL*4 vs. *8
Warning #543 nefmks.f,2966: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3095: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3097: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3106: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3108: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3110: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3112: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3115: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3123: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3162: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3165: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3248: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3349: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3351: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3389: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3397: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3403: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3412: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3598: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3605: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3792: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3928: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,3929: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,4070: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,4216: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,4280: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,4282: loss of precision in assignment: REAL*8 to REAL*4
Error #670 nefmks.f,5699: intrinsic MAX() args differ - REAL*4 vs. *8
Error #670 nefmks.f,5711: intrinsic MAX() args differ - REAL*4 vs. *8
Warning #543 nefmks.f,6334: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6335: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6336: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6337: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6342: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6363: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6371: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6388: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6389: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6393: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6444: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6459: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6460: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6462: loss of precision in assignment: REAL*8 to REAL*4

Warning #543 nefmks.f,6463: loss of preci in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6486: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6489: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6493: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6495: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6496: loss of precision in assignment: REAL*8 to REAL*4
Warning #543 nefmks.f,6811: loss of precision in assignment: REAL*8 to REAL*4
Error #670 nefmks.f,7883: intrinsic MAX() args differ - REAL*4 vs. *8
Warning #682 nefmks.f,7896: different size reals used in arg #1 of TPPRT()
Warning #682 nefmks.f,7896: different size reals used in arg #2 of TPPRT()
Warning #682 nefmks.f,7897: different size reals used in arg #2 of TPPRT()
Warning #682 nefmks.f,7920: different size reals used in arg #2 of TPPRT()
Warning #682 nefmks.f,7922: different size reals used in arg #2 of TPPRT()
Warning #103 nefmks.f,8117: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,8117: expecting REAL constant in DATA - not REAL*8
Warning #103 nefmks.f,8126: expecting REAL constant in DATA - not REAL*8
Warning #543 nefmks.f,8127: loss of precision in assignment: REAL*8 to REAL*4

pcl5

JOB 324

nefmks.rpt.4st

1/15/98

no 4 study problems

For: cscherer
Date: Thu Jan 15 11:55:58 CST 1998
Submit queue: IF 1 / Ethernet / UHSW
Submitted: Wed Jan 15 13:21:51 1998
Started: Wed Jan 15 13:21:51 1998

Warning #700 nefmks.f,456: No interface f routine FDATE()

32

can resolve by adding interfaces
defn. to nefmk,ntf

pcl5

JOB 322

nefmks.dif

diff nefmks.f. dev

↓
has code changes
not found in 4study
version

nefmks.f. 4st

↓
has variable declared
double precision that
/dev version doesn't

For: cscherer
Date: Thu Jan 15 11:53:41 CST 1998

Submit queue: IF 1 / Ethernet / UHSW
Submitted: Wed Jan 15 13:18:39 1998
Started: Wed Jan 15 13:18:39 1998

recommend merge /dev code changes (marked
w/) into 4 study version + review for study
Cred.S.

QMS

QMS 3825 Print System

QMS 3825 Print System

```

138c138
< REAL LAMBDA, C, AW, P
---
> REAL*8 LAMBDA, C, AW, P
142c142
< REAL PEAK(MXISO), TIMBRK(MXISO), TIMPK(MXISO), TODT(MXISO),
---
> REAL*8 PEAK(MXISO), TIMBRK(MXISO), TIMPK(MXISO), TODT(MXISO),
147c147
< REAL BCJ(MXJCT)
---
> REAL*8 BCJ(MXJCT)
152c152
< REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
> REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
162c162
< REAL PORE(MXPTH), DT, Y
---
> REAL*8 PORE(MXPTH), DT, Y
169c169
< REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
> REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
337c337
< REAL LAMBDA, C, AW, P
---
> REAL*8 LAMBDA, C, AW, P
349c349
< REAL PEAK(MXISO), TODT(MXISO), WDIS(MXISO), WDISUM,
---
> REAL*8 PEAK(MXISO), TODT(MXISO), WDIS(MXISO), WDISUM,
354c354
< REAL BCJ(MXJCT)
---
> REAL*8 BCJ(MXJCT)
362c362
< REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
> REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
372c372
< REAL TH(MXLEG), Q(MXLEG), PATH(MXLEG)
---
> REAL*8 TH(MXLEG), Q(MXLEG), PATH(MXLEG)
376c376
< REAL PORE(MXPTH), DT, Y
---
> REAL*8 PORE(MXPTH), DT, Y
383c383
< REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
> REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
387c387
< REAL TSRCE(MXQSC), QSRCE(MXQSC)
---
> REAL*8 TSRCE(MXQSC), QSRCE(MXQSC)
393c393
< REAL QSC, VOL
---
> REAL*8 QSC, VOL

```


397c397

```
< REAL TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
```

```
> REAL*8 TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
```

401c401

```
< REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
```

```
> REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
```

406c406

```
< REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
```

```
> REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
```

430,434c430

<

```
< cc rwj 10-10-97 Increase FN to handle 43 chains, when generating DISCHARGE file
```

```
< cc CHARACTER FN(6)*9, TITLE*80, TTLGEN*80, DUM*1, DLST(NNINP)*20
```

```
< CHARACTER FN(43)*9, TITLE*80, TTLGEN*80, DUM*1, DLST(NNINP)*20
```

<

```
> CHARACTER FN(6)*9, TITLE*80, TTLGEN*80, DUM*1, DLST(NNINP)*20
```

443,458c439,440

<

```
< cc rwj 10-10-97 Increase FN to handle 43 chains, when generating DISCHARGE file
```

```
< cc DATA FN/ 'TAPE15' , 'TAPE16' , 'TAPE17' ,
```

```
< cc X 'TAPE18' , 'TAPE19' , 'TAPE20' /
```

```
< DATA FN/
```

```
< & 'Chain01', 'Chain02', 'Chain03', 'Chain04', 'Chain05',
```

```
< & 'Chain06', 'Chain07', 'Chain08', 'Chain09', 'Chain10',
```

```
< & 'Chain11', 'Chain12', 'Chain13', 'Chain14', 'Chain15',
```

```
< & 'Chain16', 'Chain17', 'Chain18', 'Chain19', 'Chain20',
```

```
< & 'Chain21', 'Chain22', 'Chain23', 'Chain24', 'Chain25',
```

```
< & 'Chain26', 'Chain27', 'Chain28', 'Chain29', 'Chain30',
```

```
< & 'Chain31', 'Chain32', 'Chain33', 'Chain34', 'Chain35',
```

```
< & 'Chain36', 'Chain37', 'Chain38', 'Chain39', 'Chain40',
```

```
< & 'Chain41', 'Chain42', 'Chain43'
```

```
< & /
```

<

```
> DATA FN/ 'TAPE15' , 'TAPE16' , 'TAPE17' ,
```

```
> X 'TAPE18' , 'TAPE19' , 'TAPE20' /
```

569,573c551

<

```
< cc rwj 10-10-97 Change file mode.
```

```
< cc OPEN (30, FILE=nefdis, STATUS='NEW')
```

```
< OPEN (30, FILE=nefdis, STATUS='UNKNOWN')
```

<

```
> OPEN (30, FILE=nefdis, STATUS='NEW')
```

1071,1072c1049,1050

```
< REAL MKSDX, MKSDXX, LAMBDA, TDNM
```

```
< REAL C0, C1, C2, D1, D2, D3, V1, V2, V3, DRPJ, CN, FMRPJ,
```

```
> REAL*8 MKSDX, MKSDXX, LAMBDA, TDNM
```

```
> REAL*8 C0, C1, C2, D1, D2, D3, V1, V2, V3, DRPJ, CN, FMRPJ,
```

1076c1054

```
< REAL TH(MXLEG), Q(MXLEG), PATH(MXLEG)
```

```
> REAL*8 TH(MXLEG), Q(MXLEG), PATH(MXLEG)
```

1083c1061

```
< REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
```

```

---
> REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
1091c1069
< REAL PORE(MXPTH), DT, Y
---
> REAL*8 PORE(MXPTH), DT, Y
1098c1076
< REAL VOL, QSC
---
> REAL*8 VOL, QSC
1102c1080
< REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
---
> REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
1107c1085
< REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
> REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
1114c1092
< REAL GA(MXNVI), R, SIGIR, TNUM, VL(MXMEM,MXNVI)
---
> REAL*8 GA(MXNVI), R, SIGIR, TNUM, VL(MXMEM,MXNVI)
1480,1481c1458,1459
< REAL LAMBDA
< REAL DL, DM, TSI, DFR, DLIN, DR, DTIN, DSTSAV, SUM,
---
> REAL*8 LAMBDA
> REAL*8 DL, DM, TSI, DFR, DLIN, DR, DTIN, DSTSAV, SUM,
1484c1462
< REAL ABFX
---
> REAL*8 ABFX
1487c1465
< REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
> REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
1495c1473
< REAL TH(MXLEG), PATH(MXLEG), Q(MXLEG)
---
> REAL*8 TH(MXLEG), PATH(MXLEG), Q(MXLEG)
1499c1477
< REAL PORE(MXPTH), DT, Y
---
> REAL*8 PORE(MXPTH), DT, Y
1506c1484
< REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
---
> REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
1511c1489
< REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
> REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
1518c1496
< REAL TS(MXMEM), VL(MXMEM,MXNVI), VA(MXMEM,MXNVI),
---
> REAL*8 TS(MXMEM), VL(MXMEM,MXNVI), VA(MXMEM,MXNVI),
1942,1943c1920,1921
< REAL DIS, DSMX, DTEF, DKR, DXLG, DTE, FKR, FKR1, PV
< REAL TS(MXMEM), VA(MXMEM,MXNVI), WF
---

```

```

> REAL*8 DIS, DSMX, DTEF, DKR, DXLG E, FKR, FKR1, PV
> REAL*8 TS(MXMEM), VA(MXMEM,MXNVI), WF
1945c1923
< REAL ABFX
---
> REAL*8 ABFX
1948c1926
< REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
---
> REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
1953c1931
< REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
> REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
2077,2079c2055,2057
< REAL LAMBDA
< REAL RMNT, T1, T2, AA, AY, FMNT, DT1, DT2, DT3, DT4, RNT
< REAL DTSIG, DEPLEN, TM, FCTR, FNT, SIGT, CM
---
> REAL*8 LAMBDA
> REAL*8 RMNT, T1, T2, AA, AY, FMNT, DT1, DT2, DT3, DT4, RNT
> REAL*8 DTSIG, DEPLEN, TM, FCTR, FNT, SIGT, CM
2082,2083c2060,2061
< cc REAL DXK(MXMEM), DTMIN, DTX, CNX, VDT(MXPTH)
< REAL DXK(MXPTH), DTMIN, DTX, CNX, VDT(MXPTH)
---
> cc REAL*8 DXK(MXMEM), DTMIN, DTX, CNX, VDT(MXPTH)
> REAL*8 DXK(MXPTH), DTMIN, DTX, CNX, VDT(MXPTH)
2087c2065
< REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
> REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
2095c2073
< REAL TH(MXLEG), PATH(MXLEG), Q(MXLEG)
---
> REAL*8 TH(MXLEG), PATH(MXLEG), Q(MXLEG)
2099c2077
< REAL Y, DT, PORE(MXPTH)
---
> REAL*8 Y, DT, PORE(MXPTH)
2106c2084
< REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM),
---
> REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM),
2111c2089
< REAL VOL, QSC
---
> REAL*8 VOL, QSC
2115c2093
< REAL TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
---
> REAL*8 TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
2119c2097
< REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
---
> REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
2124c2102
< REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
> REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),

```

2637,2638c2615,2616

```
< REAL LAMBDA
< REAL V1, VDT, VDTK, DT, Y
---
```

```
> REAL*8 LAMBDA
> REAL*8 V1, VDT, VDTK, DT, Y
```

2641c2619

```
< REAL TH(MXLEG), Q(MXLEG), PATH(MXLEG)
---
```

```
> REAL*8 TH(MXLEG), Q(MXLEG), PATH(MXLEG)
```

2645c2623

```
< REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
```

```
> REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
```

2653c2631

```
< REAL PORE(MXPTH)
---
```

```
> REAL*8 PORE(MXPTH)
```

2668d2645

```
< cc rwj 9-3-97 Add **2 for veliso per T. McCartin.
```

2670c2647

```
< & (-GA)*SQRT(2.0*ALPHA(K)*VELISO(K,IR)**2/path(k))
---
```

```
> & (-GA)*SQRT(2.0*ALPHA(K)*VELISO(K,IR)/path(k))
```

2684,2685c2661,2662

```
< REAL LAMBDA, D, EX1, EX2, TDNX
< REAL FCTR, SUM, ANS, SFMX, CTF, XM, VR, VP, DT, BMN,
---
```

```
> REAL*8 LAMBDA, D, EX1, EX2, TDNX
> REAL*8 FCTR, SUM, ANS, SFMX, CTF, XM, VR, VP, DT, BMN,
```

2689c2666

```
< REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
```

```
> REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
```

2697c2674

```
< REAL PORE(MXPTH)
---
```

```
> REAL*8 PORE(MXPTH)
```

2704c2681

```
< REAL F(NDDF), RHO(MXGRD,MXMEM), RHONEW(MXGRD),
---
```

```
> REAL*8 F(NDDF), RHO(MXGRD,MXMEM), RHONEW(MXGRD),
```

2708c2685

```
< REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
```

```
> REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
```

3081,3084c3058,3061

```
< REAL B1, B2, T1, T2, DT, S1, S2
< REAL D(*),E1,E2,DN1,DN2,R1,R2,SUM,P
< REAL T11,T12,T21,T22,T11T,T12T,T21T,T22T
< REAL ANS, CJ, D1, D2
---
```

```
> REAL*8 B1, B2, T1, T2, DT, S1, S2
> REAL*8 D(*),E1,E2,DN1,DN2,R1,R2,SUM,P
> REAL*8 T11,T12,T21,T22,T11T,T12T,T21T,T22T
> REAL*8 ANS, CJ, D1, D2
```

3141c3118

```
< REAL FUNCTION PRP(D,B,V,VP,DL,N,T1,T2)
---
```

```
> REAL*8 FUNCTION PRP(D,B,V,VP,DL,N,T1,T2)
```

3145,3146c3122,3123

```
< REAL D(*),F1,F2,DENOM,DF,P1,P2,P3,X
< REAL PROD, ZZ, T1, T2, B(*), V, VP, DL
---
> REAL*8 D(*),F1,F2,DENOM,DF,P1,P2,P3,X
> REAL*8 PROD, ZZ, T1, T2, B(*), V, VP, DL
```

3184,3185c3161,3162

```
< REAL LAMBDA, DSV
< REAL DTLT, DTLT2, DTSM, SUM, TDP
---
> REAL*8 LAMBDA, DSV
> REAL*8 DTLT, DTLT2, DTSM, SUM, TDP
```

3187c3164

```
< REAL TSPFAC
---
> REAL*8 TSPFAC
```

3189c3166

```
< REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
> REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
```

3197c3174

```
< REAL PORE(MXPTH), DT, Y
---
> REAL*8 PORE(MXPTH), DT, Y
```

3204c3181

```
< REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
---
> REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
```

3209c3186

```
< REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
> REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
```

3216c3193

```
< REAL BFR(MXMEM), D(MXMEM), DS(MXMEM)
---
> REAL*8 BFR(MXMEM), D(MXMEM), DS(MXMEM)
```

3333c3310

```
< REAL FUNCTION TSPFAC(DT,D,B,LT,TDP)
---
> REAL*8 FUNCTION TSPFAC(DT,D,B,LT,TDP)
```

3344,3346c3321,3323

```
< REAL D(*),DC(5),DN(9),A,SUM,SUM1,PM,P
< REAL F2,DCL,R12,R23,R24,R35, TDP
< REAL SUMS, DT, B(*)
---
> REAL*8 D(*),DC(5),DN(9),A,SUM,SUM1,PM,P
> REAL*8 F2,DCL,R12,R23,R24,R35, TDP
> REAL*8 SUMS, DT, B(*)
```

3451c3428

```
< REAL C, AW, P
---
> REAL*8 C, AW, P
```

3453c3430

```
< REAL BCJ(MXJCT)
---
> REAL*8 BCJ(MXJCT)
```

3460c3437

```
< REAL PATH(MXLEG), Q(MXLEG), TH(MXLEG)
---
> REAL*8 PATH(MXLEG), Q(MXLEG), TH(MXLEG)
```

3464c3441

< REAL PORE(MXPTH), DT, Y

> REAL*8 PORE(MXPTH), DT, Y

3678c3655

< REAL C, AW, AM, SUM

> REAL*8 C, AW, AM, SUM

3680c3657

< REAL BCJ(MXJCT)

> REAL*8 BCJ(MXJCT)

3752,3754c3729,3731

< REAL MKSP, MKSQ, MKSPOR, MKSQSC, MKSQDI, MKSY, PDIS, TT,

< 1 MKSPDI

< REAL C, AW, P, PDF

> REAL*8 MKSP, MKSQ, MKSPOR, MKSQSC, MKSQDI, MKSY, PDIS, TT,

> 1 MKSPDI

> REAL*8 C, AW, P, PDF

3756c3733

< REAL BCJ(MXJCT)

> REAL*8 BCJ(MXJCT)

3763c3740

< REAL PATH(MXLEG), Q(MXLEG), TH(MXLEG)

> REAL*8 PATH(MXLEG), Q(MXLEG), TH(MXLEG)

3767c3744

< REAL PORE(MXPTH), DT, Y

> REAL*8 PORE(MXPTH), DT, Y

3774c3751

< REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS

> REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS

3778c3755

< REAL VOL, QSC

> REAL*8 VOL, QSC

3782c3759

< REAL TDTM(MXTDV), TDSAT(MXTDV,MXPTH), TDVEL(MXTDV,MXPTH),

> REAL*8 TDTM(MXTDV), TDSAT(MXTDV,MXPTH), TDVEL(MXTDV,MXPTH),

3973c3950

< REAL C, AW

> REAL*8 C, AW

3977c3954

< REAL BCJ(MXJCT)

> REAL*8 BCJ(MXJCT)

3982c3959

< REAL TH(MXLEG), PATH(MXLEG), Q(MXLEG)

> REAL*8 TH(MXLEG), PATH(MXLEG), Q(MXLEG)

3986c3963

< REAL PORE(MXPTH), DT, Y

> REAL*8 PORE(MXPTH), DT, Y

```

4039c4016
< REAL T1, T2, DT, TUP, Y, DLG, DLT, DIS, DISI, PDIS,
---
> REAL*8 T1, T2, DT, TUP, Y, DLG, DLT, DIS, DISI, PDIS,
4042c4019
< REAL PATH(MXLEG), Q(MXLEG), TH(MXLEG)
---
> REAL*8 PATH(MXLEG), Q(MXLEG), TH(MXLEG)
4046c4023
< REAL PORE(MXPETH)
---
> REAL*8 PORE(MXPETH)
4053c4030
< REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
> REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
4060c4037
< REAL TDTM(MXTDV), TDSAT(MXTDV,MXPETH), TDVEL(MXTDV,MXPETH),
---
> REAL*8 TDTM(MXTDV), TDSAT(MXTDV,MXPETH), TDVEL(MXTDV,MXPETH),
4137,4139c4114,4116
< REAL DFF, EA, SUM, FMAX, T1, T2, TME, BLKS, TM1BLK
< REAL TTRL, T3, DL, VR, VRP, TMAX, D(MXMEM), BFR(MXMEM)
< REAL FM, TLED, TMIN, LAMBDA, DF1, RATF1
---
> REAL*8 DFF, EA, SUM, FMAX, T1, T2, TME, BLKS, TM1BLK
> REAL*8 TTRL, T3, DL, VR, VRP, TMAX, D(MXMEM), BFR(MXMEM)
> REAL*8 FM, TLED, TMIN, LAMBDA, DF1, RATF1
4141c4118
< REAL ET, PRP
---
> REAL*8 ET, PRP
4143c4120
< REAL CUROUT(MXMEM), VELISO(MXPETH,MXMEM), TRAVT(MXISO),
---
> REAL*8 CUROUT(MXMEM), VELISO(MXPETH,MXMEM), TRAVT(MXISO),
4151c4128
< REAL PORE(MXPETH), DT, Y
---
> REAL*8 PORE(MXPETH), DT, Y
4158c4135
< REAL DX(MXPETH), TMN(MXTSP), BF(NDBF)
---
> REAL*8 DX(MXPETH), TMN(MXTSP), BF(NDBF)
4163c4140
< REAL F(NDDF), SF(NDSF), RHO(MXGRD,MXMEM), RHONEW(MXGRD),
---
> REAL*8 F(NDDF), SF(NDSF), RHO(MXGRD,MXMEM), RHONEW(MXGRD),
4167c4144
< REAL V(MXNVI,MXSUB,MXPETH), DXX, B(2*MXNVI, MXSUB, MXPETH),
---
> REAL*8 V(MXNVI,MXSUB,MXPETH), DXX, B(2*MXNVI, MXSUB, MXPETH),
4195,4197d4171
< crbc92297
< if(l.gt.ntx) l=ntx
< crbc
4412,4413c4386,4387
< REAL MKSPETH, MKSARE, MKSCON
< REAL LAMBDA, C, AW, P
---

```

```

> REAL*8 MKSPTH, MKSARE, MKSCON
> REAL*8 LAMBDA, C, AW, P
4415c4389
< REAL TIMBRK(MXISO), TIMPK(MXISO), PEAK(MXISO), WDIS(MXISO),
---
> REAL*8 TIMBRK(MXISO), TIMPK(MXISO), PEAK(MXISO), WDIS(MXISO),
4420c4394
< REAL BCJ(MXJCT)
---
> REAL*8 BCJ(MXJCT)
4425c4399
< REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
> REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
4435c4409
< REAL TH(MXLEG), PATH(MXLEG), Q(MXLEG)
---
> REAL*8 TH(MXLEG), PATH(MXLEG), Q(MXLEG)
4439c4413
< REAL PORE(MXPTH), DT, Y
---
> REAL*8 PORE(MXPTH), DT, Y
4446c4420
< REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
> REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
4450c4424
< REAL TSRCE(MXQSC), QSRCE(MXQSC)
---
> REAL*8 TSRCE(MXQSC), QSRCE(MXQSC)
4456c4430
< REAL VOL, QSC
---
> REAL*8 VOL, QSC
4460c4434
< REAL TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
---
> REAL*8 TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
4464c4438
< REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
---
> REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
5332,5333c5306,5307
< REAL LAMBDA, DTSAVE, PRF
< REAL SOLG, FLCH, T, TDT, SUMA, SUMC, SUMU, TRAT, TW, TOTG,
---
> REAL*8 LAMBDA, DTSAVE, PRF
> REAL*8 SOLG, FLCH, T, TDT, SUMA, SUMC, SUMU, TRAT, TW, TOTG,
5341c5315
< REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
> REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
5349c5323
< REAL PORE(MXPTH), DT, Y
---
> REAL*8 PORE(MXPTH), DT, Y
5356c5330
< REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
> REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS

```


5360c5334

< REAL TSRCE(MXQSC), QSRCE(MXQSC)

> REAL*8 TSRCE(MXQSC), QSRCE(MXQSC)

5363c5337

< REAL VOL, QSC

> REAL*8 VOL, QSC

5367c5341

< REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),

> REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),

5374c5348

< REAL UNT(MXISO), AT(MXISO), PF(MXSUB,MXCHNS)

> REAL*8 UNT(MXISO), AT(MXISO), PF(MXSUB,MXCHNS)

5796c5770

< REAL LAMBDA, T1, T2, TT, PDIS

> REAL*8 LAMBDA, T1, T2, TT, PDIS

5801c5775

< REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),

> REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),

5809c5783

< REAL Q(MXLEG), PATH(MXLEG), TH(MXLEG)

> REAL*8 Q(MXLEG), PATH(MXLEG), TH(MXLEG)

5813c5787

< REAL PORE(MXPTH), DT, Y

> REAL*8 PORE(MXPTH), DT, Y

5820c5794

< REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS

> REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS

5827c5801

< REAL TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),

> REAL*8 TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),

5831c5805

< REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)

> REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)

5836c5810

< REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),

> REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),

5843c5817

< REAL VDT(MXPTH), DXSV(MXPTH), VDTSV(MXPTH)

> REAL*8 VDT(MXPTH), DXSV(MXPTH), VDTSV(MXPTH)

6068c6042

< REAL LAMBDA, ET

> REAL*8 LAMBDA, ET

6070c6044

< REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),

> REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),

6078c6052

< REAL PORE(MXPTH), DT, Y

> REAL*8 PORE(MXPTH), DT, Y

6083c6057

< REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),

> REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),

6098,6099c6072,6073

< REAL RM(MXMEM), DMIN, RMX, BFR(MXMEM)

< REAL D(MXMEM)

> REAL*8 RM(MXMEM), DMIN, RMX, BFR(MXMEM)

> REAL*8 D(MXMEM)

6303c6277

< REAL FUNCTION ET(T,IP,D,B)

> REAL*8 FUNCTION ET(T,IP,D,B)

6305,6306c6279,6280

< REAL D(*), DC(5), DN(5)

< REAL PROD, PM, F2, A, SUM, SUM1, T, B(*)

> REAL*8 D(*), DC(5), DN(5)

> REAL*8 PROD, PM, F2, A, SUM, SUM1, T, B(*)

6408,6409c6382,6383

< REAL TIME, DJ, DI, RAT

< REAL T,A1,A2,ARG1,ARG2,A,ARG,ELF,ADD,S,TS,TSS

> REAL*8 TIME, DJ, DI, RAT

> REAL*8 T,A1,A2,ARG1,ARG2,A,ARG,ELF,ADD,S,TS,TSS

6512c6486

< REAL RESIDE, DT1, DT2, DT3, DT4, DTH, TTMIN, QSKQ, QMAX,

> REAL*8 RESIDE, DT1, DT2, DT3, DT4, DTH, TTMIN, QSKQ, QMAX,

6514c6488

< REAL LAMBDA, D, PROD, DEC

> REAL*8 LAMBDA, D, PROD, DEC

6516c6490

< REAL ET

> REAL*8 ET

6518c6492

< REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),

> REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),

6526c6500

< REAL PORE(MXPTH), DT, Y

> REAL*8 PORE(MXPTH), DT, Y

6533c6507

< REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS

> REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS

6537c6511

< REAL TSRCE(MXQSC), QSRCE(MXQSC)

> REAL*8 TSRCE(MXQSC), QSRCE(MXQSC)

6540c6514

< REAL QSC, VOL

```

---
> REAL*8 QSC, VOL
6544c6518
< REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
> REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
6551c6525
< REAL BFR(MXISO), AT(MXISO), PF(MXSUB,MXCHNS)
---
> REAL*8 BFR(MXISO), AT(MXISO), PF(MXSUB,MXCHNS)
6795c6769
< REAL QSC, QDIS, VOL, T1, T2, F, F1, TSTOP, ARG
---
> REAL*8 QSC, QDIS, VOL, T1, T2, F, F1, TSTOP, ARG
6797c6771
< REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM)
---
> REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM)
6846c6820
< REAL LAMBDA
---
> REAL*8 LAMBDA
6859c6833
< REAL TS, T1, TRAT, DELT, T
---
> REAL*8 TS, T1, TRAT, DELT, T
6863c6837
< REAL PEAK(MXISO), TOTD(MXISO), WDIS(MXISO), WDISUM,
---
> REAL*8 PEAK(MXISO), TOTD(MXISO), WDIS(MXISO), WDISUM,
6871c6845
< REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
> REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
6881c6855
< REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
> REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
6886c6860
< REAL TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
---
> REAL*8 TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
6891c6865
< REAL TSCL(NTMX), RATE(MXISO), RATEL(MXISO), RATER(MXISO),
---
> REAL*8 TSCL(NTMX), RATE(MXISO), RATEL(MXISO), RATER(MXISO),
7191,7196c7165
<
< cc rwj 10-17-97 Make format the same as unit 6. [IOPT(13)=-1]
< cc WRITE(30,*) T, (RATE(K),K=1,NICH)
< WRITE(30,9002) T, (RATE(K),K=1,NICH)
< 9002 FORMAT(1X,1PE10.4,10(1X,1PE11.4))
<
---
> WRITE(30,*) T, (RATE(K),K=1,NICH)
7217c7186
< REAL X(*), A
---
> REAL*8 X(*), A
7244c7213

```

```

<      REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
>      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
7286c7255
<      REAL PORE(MXPPTH), DT, Y
---
>      REAL*8 PORE(MXPPTH), DT, Y
7291c7260
<      REAL QSC, VOL
---
>      REAL*8 QSC, VOL
7421,7423c7390,7392
<      REAL LAMBDA, C, AW, P
<      REAL MKSQSA, MKSSAL, MKSQDA, MKSARE, MKSALP, MKSDX, MKSP
<      REAL MKSCON, MKSPOR, MKSEL, MKSTDV, MKSVOL, MKSPTH, MKSQSR
---
>      REAL*8 LAMBDA, C, AW, P
>      REAL*8 MKSQSA, MKSSAL, MKSQDA, MKSARE, MKSALP, MKSDX, MKSP
>      REAL*8 MKSCON, MKSPOR, MKSEL, MKSTDV, MKSVOL, MKSPTH, MKSQSR
7425,7426c7394,7395
<      REAL PEAK(MXISO), TIMBRK(MXISO), TIMPK(MXISO), TOTD(MXISO)
<      REAL WDIS(MXISO), WDISUM
---
>      REAL*8 PEAK(MXISO), TIMBRK(MXISO), TIMPK(MXISO), TOTD(MXISO)
>      REAL*8 WDIS(MXISO), WDISUM
7430c7399
<      REAL BCJ(MXJCT)
---
>      REAL*8 BCJ(MXJCT)
7438c7407
<      REAL CUROUT(MXMEM), VELISO(MXPPTH,MXMEM), TRAVT(MXISO),
---
>      REAL*8 CUROUT(MXMEM), VELISO(MXPPTH,MXMEM), TRAVT(MXISO),
7448c7417
<      REAL TH(MXLEG), Q(MXLEG), PATH(MXLEG)
---
>      REAL*8 TH(MXLEG), Q(MXLEG), PATH(MXLEG)
7452c7421
<      REAL PORE(MXPPTH), DT, Y
---
>      REAL*8 PORE(MXPPTH), DT, Y
7459c7428
<      REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
>      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
7463c7432
<      REAL TSRCE(MXQSC), QSRCE(MXQSC)
---
>      REAL*8 TSRCE(MXQSC), QSRCE(MXQSC)
7469c7438
<      REAL QSC, VOL
---
>      REAL*8 QSC, VOL
7473c7442
<      REAL TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPPTH),
---
>      REAL*8 TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPPTH),
7477c7446
<      REAL DX(MXPPTH), TMN(MXTSP), BF(NDBF)
---

```

```

> REAL*8 DX(MXPTH), TMN(MXTSP), BF(I )
7482c7451
< REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
> REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
7800,7801c7769,7770
< REAL CHK, TM, T, SIGT, TL, VV, TSTART, TEND, T1, T2, TSUM
< REAL LAMBDA
---
> REAL*8 CHK, TM, T, SIGT, TL, VV, TSTART, TEND, T1, T2, TSUM
> REAL*8 LAMBDA
7803c7772
< REAL CROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
> REAL*8 CROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
7811c7780
< REAL PORE(MXPTH), DT, Y
---
> REAL*8 PORE(MXPTH), DT, Y
7818c7787
< REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
> REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
7825c7794
< REAL QSC, VOL
---
> REAL*8 QSC, VOL
7829c7798
< REAL TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
---
> REAL*8 TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV,MXPTH),
7833c7802
< REAL F(NDDF), RHOM(MXGRD,MXMEM), RHONEW(MXGRD), SF(NDSF),
---
> REAL*8 F(NDDF), RHOM(MXGRD,MXMEM), RHONEW(MXGRD), SF(NDSF),
7844,7847c7813,7816
< REAL curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
< REAL dr2tim(maxrec), dr2vol(maxrec)
< REAL filcur(mxiso,maxrec), filtim(maxrec)
< REAL filvol(maxrec), tarr(maxtim), totvol(maxtim)
---
> REAL*8 curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
> REAL*8 dr2tim(maxrec), dr2vol(maxrec)
> REAL*8 filcur(mxiso,maxrec), filtim(maxrec)
> REAL*8 filvol(maxrec), tarr(maxtim), totvol(maxtim)
8010,8011c7979,7980
< REAL LAMBDA, RAT12, RAT23, RAT13, GIT
< REAL E1, E2, E3, B, C, GPART, A, V, T, TL, TSET,
---
> REAL*8 LAMBDA, RAT12, RAT23, RAT13, GIT
> REAL*8 E1, E2, E3, B, C, GPART, A, V, T, TL, TSET,
8020c7989
< REAL CROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
> REAL*8 CROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
8028c7997
< REAL PORE(MXPTH), DT, Y
---
> REAL*8 PORE(MXPTH), DT, Y
8035c8004

```

```

<      REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
>      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
8039c8008
<      REAL QSC, VOL
---
>      REAL*8 QSC, VOL
8109c8078
<      REAL FUNCTION GIT(T,X,V,AL)
---
>      REAL*8 FUNCTION GIT(T,X,V,AL)
8111,8112c8080,8081
<      REAL A(5), P, TRPI, T, X, V, AL, ANS, B, Z
<      REAL PLYET1, PLYET2, ET1, ET2, ZMB, TERM1, TERM2, XX, XXEX
---
>      REAL*8 A(5), P, TRPI, T, X, V, AL, ANS, B, Z
>      REAL*8 PLYET1, PLYET2, ET1, ET2, ZMB, TERM1, TERM2, XX, XXEX
8171c8140
<      REAL T, MKSQDI, DT2, DTFP, LAMBDA, TM, TSET
---
>      REAL*8 T, MKSQDI, DT2, DTFP, LAMBDA, TM, TSET
8173,8176c8142,8145
<      REAL totvol(maxtim)
<      REAL curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
<      REAL dr2tim(maxrec), dr2vol(maxrec), filtim(maxrec)
<      REAL filcur(mxiso,maxrec), filvol(maxrec), tarr(maxtim)
---
>      REAL*8 totvol(maxtim)
>      REAL*8 curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
>      REAL*8 dr2tim(maxrec), dr2vol(maxrec), filtim(maxrec)
>      REAL*8 filcur(mxiso,maxrec), filvol(maxrec), tarr(maxtim)
8180c8149
<      REAL PEAK(MXISO), TOTD(MXISO), WDIS(MXISO), WDISUM,
---
>      REAL*8 PEAK(MXISO), TOTD(MXISO), WDIS(MXISO), WDISUM,
8185c8154
<      REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
>      REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
8193c8162
<      REAL PORE(MXPTH), DT, Y
---
>      REAL*8 PORE(MXPTH), DT, Y
8200c8169
<      REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
>      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
8305,8306c8274,8275
<      REAL LAMBDA, T, SUM, SUM1, SUM2, SUM3, SUM4, SUM5, SUM6
<      REAL TX1, TATM, TX2, SR1, SR2, TS1, TS2, S24, TSET, BM
---
>      REAL*8 LAMBDA, T, SUM, SUM1, SUM2, SUM3, SUM4, SUM5, SUM6
>      REAL*8 TX1, TATM, TX2, SR1, SR2, TS1, TS2, S24, TSET, BM
8308c8277
<      REAL ABFX
---
>      REAL*8 ABFX
8314c8283
<      REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---

```

```

>      REAL*8 CUROUT(MXMEM), VELISO(MXPTH, MEM), TRAVT(MXISO),
3322c8291
<      REAL PORE(MXPTH), DT, Y
---
>      REAL*8 PORE(MXPTH), DT, Y
8329c8298
<      REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
>      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
8333c8302
<      REAL QSC, VOL
---
>      REAL*8 QSC, VOL
8337c8306
<      REAL TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV, MXPTH),
---
>      REAL*8 TDTM(MXTDV), TDDT(MXTDV), TDSAT(MXTDV, MXPTH),
8341c8310
<      REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
---
>      REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
8346c8315
<      REAL F(NDDF), SF(NDSF), RHO(MXGRD, MXMEM), RHONEW(MXGRD),
---
>      REAL*8 F(NDDF), SF(NDSF), RHO(MXGRD, MXMEM), RHONEW(MXGRD),
8350c8319
<      REAL V(MXNVI, MXSUB, MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
>      REAL*8 V(MXNVI, MXSUB, MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
8445,8447c8414
< cc rwj 9-2-97 recind change for test.
< c      KNTIP = Je - IR
<      KNTIP = Js - IR
---
>      KNTIP = Je - IR
8601,8602c8568,8569
<      REAL LAMBDA, TX1, TX2, TS1, TS2, SR1, SR2, S24
<      REAL S, SUM, TP, TRAT, TRAT2, RT, RT2
---
>      REAL*8 LAMBDA, TX1, TX2, TS1, TS2, SR1, SR2, S24
>      REAL*8 S, SUM, TP, TRAT, TRAT2, RT, RT2
8604c8571
<      REAL CUROUT(MXMEM), VELISO(MXPTH, MXMEM), TRAVT(MXISO),
---
>      REAL*8 CUROUT(MXMEM), VELISO(MXPTH, MXMEM), TRAVT(MXISO),
8612c8579
<      REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
>      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
8616c8583
<      REAL QSC, VOL
---
>      REAL*8 QSC, VOL
8620c8587
<      REAL V(MXNVI, MXSUB, MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
>      REAL*8 V(MXNVI, MXSUB, MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
8627c8594
<      REAL F(NDDF), SF(NDSF), RHO(MXGRD, MXMEM), RHONEW(MXGRD),
---

```

```

>      REAL*8 F(NDDF), SF(NDSF), RHO(MXG: (MEM), RHONEW(MXGRD),
8769c8736
<      REAL LAMBDA, CF, TA, PC, CM, DIF, DPI
---
>      REAL*8 LAMBDA, CF, TA, PC, CM, DIF, DPI
8771c8738
<      REAL CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
---
>      REAL*8 CUROUT(MXMEM), VELISO(MXPTH,MXMEM), TRAVT(MXISO),
8779c8746
<      REAL PORE(MXPTH), DT, Y
---
>      REAL*8 PORE(MXPTH), DT, Y
8787c8754
<      REAL DX(MXPTH), TMN(MXTSP), BF(NDBF)
---
>      REAL*8 DX(MXPTH), TMN(MXTSP), BF(NDBF)
8791c8758
<      REAL F(NDDF), SF(NDSF), RHO(MXGRD,MXMEM), RHOM(MXGRD,MXMEM),
---
>      REAL*8 F(NDDF), SF(NDSF), RHO(MXGRD,MXMEM), RHOM(MXGRD,MXMEM),
8795c8762
<      REAL V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
---
>      REAL*8 V(MXNVI,MXSUB,MXPTH), DXX, B(2*MXNVI, MXSUB, MXPTH),
8911c8878
<      REAL dummyarray(maxtim,1), LAMBDA, tempc(mxiso)
---
>      REAL*8 dummyarray(maxtim,1), LAMBDA, tempc(mxiso)
8913,8916c8880,8883
<      REAL curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
<      REAL dr2tim(maxrec), dr2vol(maxrec)
<      REAL filcur(mxiso,maxrec), filtim(maxrec)
<      REAL filvol(maxrec), tarr(maxtim), totvol(maxtim)
---
>      REAL*8 curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
>      REAL*8 dr2tim(maxrec), dr2vol(maxrec)
>      REAL*8 filcur(mxiso,maxrec), filtim(maxrec)
>      REAL*8 filvol(maxrec), tarr(maxtim), totvol(maxtim)
8921c8888
<      REAL CUROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
---
>      REAL*8 CUROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
8929c8896
<      REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
>      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
8985c8952
<      REAL dummyarray(maxtim,1), LAMBDA, tempc(mxiso), tempv(maxtim)
---
>      REAL*8 dummyarray(maxtim,1), LAMBDA, tempc(mxiso), tempv(maxtim)
8987c8954
<      REAL curies(maxtim,mxiso), dr2tim(maxrec), filvol(maxrec),
---
>      REAL*8 curies(maxtim,mxiso), dr2tim(maxrec), filvol(maxrec),
8993c8960
<      REAL CUROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
---
>      REAL*8 CUROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
9001c8968

```



```

<      REAL TUB, TRLSE, TLCH, SUMIN(MXME.  SUMOUT(MXMEM), QDIS
---
>      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
9037c9004
<      REAL LAMBDA
---
>      REAL*8 LAMBDA
9039,9042c9006,9009
<      REAL curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
<      REAL dr2tim(maxrec), dr2vol(maxrec)
<      REAL filcur(mxiso,maxrec), filtim(maxrec)
<      REAL filvol(maxrec), tarr(maxtim), totvol(maxtim)
---
>      REAL*8 curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
>      REAL*8 dr2tim(maxrec), dr2vol(maxrec)
>      REAL*8 filcur(mxiso,maxrec), filtim(maxrec)
>      REAL*8 filvol(maxrec), tarr(maxtim), totvol(maxtim)
9047c9014
<      REAL CROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
---
>      REAL*8 CROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
9055c9022
<      REAL TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
---
>      REAL*8 TUB, TRLSE, TLCH, SUMIN(MXMEM), SUMOUT(MXMEM), QDIS
9093c9060
<      real    slope, t(maxtim), delt, dt, var(maxtim,nmax), val(nmax)
---
>      real*8    slope, t(maxtim), delt, dt, var(maxtim,nmax), val(nmax)
9140c9107
<      REAL LAMBDA
---
>      REAL*8 LAMBDA
9149c9116
<      REAL  conc, u234(maxrec), u234tm(maxrec)
---
>      REAL*8  conc, u234(maxrec), u234tm(maxrec)
9151,9154c9118,9121
<      REAL curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
<      REAL dr2tim(maxrec), dr2vol(maxrec)
<      REAL filcur(mxiso,maxrec), filtim(maxrec)
<      REAL filvol(maxrec), tarr(maxtim), totvol(maxtim)
---
>      REAL*8 curies(maxtim,mxiso), dr2cur(mxiso,maxrec)
>      REAL*8 dr2tim(maxrec), dr2vol(maxrec)
>      REAL*8 filcur(mxiso,maxrec), filtim(maxrec)
>      REAL*8 filvol(maxrec), tarr(maxtim), totvol(maxtim)
9162c9129
<      REAL CROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
---
>      REAL*8 CROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),
9236c9203
<      REAL LAMBDA
---
>      REAL*8 LAMBDA
9245c9212
<      REAL  conc
---
>      REAL*8  conc
9247c9214

```

```
<      REAL CUROUT(MXMEM), CURZTR(MXISO),  IVT(MXISO),  
---  
>      REAL*8 CUROUT(MXMEM), CURZTR(MXISO), TRAVT(MXISO),  
9258c9225  
<      REAL curies(maxtim,mxiso), dr2cur(mxiso,maxrec), tarr(maxtim),  
---  
>      REAL*8 curies(maxtim,mxiso), dr2cur(mxiso,maxrec), tarr(maxtim),
```

TPA V 3.1.3

SOFTWARE RELEASE NOTICE

| | | |
|--|----------------------|--------------------------|
| 1. SRN Number: PA-SRN-208 | | |
| 2. Project Title: TSPA & Technical Integration Code | | Project No. 20-01402-762 |
| 3. SRN Title: TPA Version 3.1.3 12/10-4-44 | | |
| 4. Originator/Requestor: Ron Janetzke | | Date: Oct. 1, 1999 |
| 5. Summary of Actions <ul style="list-style-type: none"> <input type="checkbox"/> Release of new software <input type="checkbox"/> Release of modified software: <ul style="list-style-type: none"> <input type="checkbox"/> Enhancements made <input type="checkbox"/> Corrections made <input type="checkbox"/> Change of access software <input checked="" type="checkbox"/> Software Retirement | | |
| 6. Persons Authorized Access | | |
| Name | Read Only/Read-Write | Addition/Change/Delete |
| Ron Janetzke Sitakanta Mohanty Tim McCartin (NRC) M. Rose Byrne (NRC) | RW RW RW RW | |
| 7. Element Manager Approval: <i>Golden Wittmeyer</i> | | Date: 10/4/99 |
| 8. Remarks: Version superceded. | | |

SOFTWARE SUMMARY FORM

| | | | |
|--|---|---|--|
| 01. Summary Date: 02/19/98 | 02. Summary prepared by (Name and phone) B. Mabrito (210)522-5149 | 03. Summary Action: Modified | |
| 04. Software Date: 02/19/98 | 05. Short Title: TPA Version 3.1.3 | | |
| 06. Software Title: TPA - System Performance Assessment Computer Code, Version 3.1.3 | | 07. Internal Software ID: None | |
| 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 type="checkbox"/> Scientific/Engineering <input type="checkbox"/> Auxiliary Analyses <input checked="" type="checkbox"/> Total System PA <input type="checkbox"/> Subsystem PA <input type="checkbox"/> Other B. Specific: | |
| 11. Submitting Organization and Address: CNWRA/SWRI 6220 Culebra Road San Antonio TX 78228 | | 12. Technical Contact(s) and Phone: Sitakanta Mohanty (210) 522-5185 | |
| 13. Narrative: The TPA Code consists of the following modules: UZFLOW, NFENV, EBSREL, UZFT, SZFT, DCAGW, FAULTO, SEISMO, VOLCANO, ASHPLUMO, ASHRMVO, DCAGS, LHS, EXEC. | | | |
| 14. Computer Platform: SUN Workstation | 15. Computer Operating System: UNIX | 16. Programming Language(s): FORTRAN | 17. Number of Source Program Statements: Approx. 32,000 lines w/o Stand Alone Codes |
| 18. Computer Memory Requirements: 45 Mb | 19. Tape Drives: None | 20. Disk/Drum Units: N/A | 21. Graphics: N/A |
| 22. Other Operational Requirements: None | | | |
| 23. Software Availability: <input checked="" type="checkbox"/> Available <input type="checkbox"/> Limited <input type="checkbox"/> In-House ONLY | | 24. Documentation Availability: <input type="checkbox"/> Available <input type="checkbox"/> Inadequate <input checked="" type="checkbox"/> In-House ONLY Including NRC | |
| Software Custodian: <u><i>B. Mabrito</i></u> Date: <u>2/19/98</u> | | | |

SOFTWARE RELEASE NOTICE

| | | |
|--|-------|-----------------------------|
| 01. SRN Number: PA-SRN-166 | | |
| 02. Project Title: TSPA & Technical Integration Code | | Project No.: 20-1402-762 |
| 03. SRN Title: TPA Version 3.1.3 | | |
| 04. Originator/Requestor: Bruce Mabrito | | Date: 02/19/98 |
| 05. Summary of Actions | | |
| <input type="checkbox"/> Release of new software | | |
| <input checked="" type="checkbox"/> Release of modified software: | | |
| <input checked="" type="checkbox"/> Enhancements made (modest changes per NRC request) | | |
| <input type="checkbox"/> Corrections made | | |
| <input type="checkbox"/> Change of access software | | |
| <input type="checkbox"/> Software Retirement | | |
| 06. Persons Authorized Access | | |
| Name | RO/RW | A/C/D |
| Sitakanta Mohanty | RW | |
| Ron Janetzke | RW | |
| Tim McCartin (NRC) | RW | |
| M. Rose Byrne (NRC) | RW | |
| 07. Element Manager Approval: <i>R. G. Braca</i> | | Date: 2/19/98 |
| 08. Remarks: | | |
| TPA Version 3.1.3 software was sent to NRC Headquarters on 02/19/98 | | |

6/13

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE: TPA Version 3.1.3

February 19, 1998

Total Performance Assessment (Scientific and Engineering Software) Version 3.1.3

NOTE: This version of the TPA Software contains very modest changes from the previous version released January 16, 1998. There are a total of only three (3) Software Problem Change Reports (SPCRs) associated with this version change.

1. Scientific Notebook Documentation: Changes to the TPA code have been made utilizing SPCR's that were generated by both the CNWRA and NRC staff working on TPA Version 3.1.3.
2. Programming Language: ANSI Standard FORTRAN 77 confirmed by the Software Custodian. See attached hard copies of TPA Version 3.1.3 pages as examples.
3. Internal Documentation: On 2/19/98, B. Mabrito reviewed portions of the TPA Version 3.1.3 scientific and engineering software on the "Bigbend" SUN Ultra 2 work station. The TPA Version 3.1.3 software is located on the "Scratchy1" server. The TPA code contained comments and a hard copy page showing representative comments is attached to this Design Verification Report. The internal documentation comments in TPA Version 3.1.3 meet the requirements of TOP-018, Section 5.4.4.
4. Software Labels and Data
 - a. Header Data and Format: TPA Version 3.1.3 header data and the format were compared against TOP-018 Section 5.4.6 and found acceptable (see several of the attached hard copy pages).
 - b. NRC Data: TPA Version 3.1.3 NRC data and the format were compared against TOP-018, Section 5.4.6, third bullet and found to be acceptable.
 - c. Source Code Header: TPA Version 3.1.3 header data was compared to TOP-018 Section 5.4.6, fourth bullet, and found to meet the requirements and are acceptable.
5. Unique Run Identification: At the top of each screen output file for TPA Version 3.1.3, a unique identifier is on the printed sheet. A page of files was printed and attached to this report as objective evidence, meeting the requirements of TOP-018, Section 5.4.5.
6. Software Analysis and Results: Software analysis tools were not run on TPA Version 3.1.3 because the modest nature of the changes to the code did not warrant the use of them. See the SPCR's for details.



2/19/98

Sitakanta Mohanty Date
CNWRA TPA Software Developer



2/19/98

Bruce Mabrito Date
CNWRA Software Custodian

Attachments/

Original to: Software Folder

cc: CNWRA Software Developer/Cognizant EM


```
bigbend:/bscr3/mohanty/A_tpa. 3.1.3 {35} !v
vi tpa.inp
bigbend:/bscr3/mohanty/A_tpa3.1.3 {36} time tpa.e
```

```
=====
exec: Welcome to TPA Version 3.1.3
Job started: Thu Feb 19 17:48:59 1998
=====
```

Specified Global Parameters:

```

      NumberOfRealizations =    5
      NumberOfSubareas =    7
      MaximumTime[yr] =   20000.
      Volcanism scenario =    0 (yes=1, no=0)
      Faulting scenario =    0 (yes=1, no=0)
      Seismic scenario =    0 (yes=1, no=0)
      Distance to Critical Group (km) = 20.0
```

```
The specified path for data = /bscr3/mohanty/A_tpa3.1.3/
The specified path for codes = /bscr3/mohanty/A_tpa3.1.3/
```

****To modify global parameters or the path, stop code execution using control-C****

```
-----
      subarea   1 of    7           realization   1 of    5
-----
```

```
exec: calling uzflow
exec: calling nfenv
exec: calling ebsfail
```

```
      ebsfail: time of WP failure =   15102.6   yr
```

```
exec: failed WPs from INITIAL   event =      8 at time =      0.0 yr
```

```
exec: failed WPs from CORROSION event =   1655 at time =  15102.6 yr
```

```
      *** failed WPs: all WPs failed ***
```

```
exec: calling ebsrel
```

```
      Highest release rates from Sub Area 1
```

```

      Tc99  4.2175E-01 [Ci/yr/SA] at  1.954E+04 yr
      Ni59  6.9899E-02 [Ci/yr/SA] at  1.909E+04 yr
      Am243  4.7797E-02 [Ci/yr/SA] at  2.000E+04 yr
      Np237  2.9004E-02 [Ci/yr/SA] at  1.909E+04 yr
```

```
c Program Name:          TPA - Total-System Performance Assessment - Code
c File Name:             ebsfail.f
c File Date:             02/19/98
c Release Version:       3.1.3
c
c Client Name:           USNRC
c                        U. S. Nuclear Regulatory Commission
c                        NRC Office of Nuclear Material Safety and Safeguards
c                        Division of Waste Management
c Contract Number:       NRC 02-93-005
c
c NRC Contact            Tim McCartin (301) 415-6681
c
c CNWRA Contact:         Sitakanta Mohanty (210) 522-5185
c                        Center for Nuclear Waste Regulatory Analyses
c                        San Antonio, Texas 78238-5166
c                        smohanty@swri.edu
c
c Documentation:         "Total-System Performance Assessment (TPA)
c                        Version 3.1 Code: Module Description and User's
c                        Guide", Center for Nuclear Waste Regulatory
c                        Analyses (in preparation)
```

[illegible]

c

C
C D I S C L A I M E R
C
C = = = = =

"This computer code/material was prepared as an account of work performed by the Center for Nuclear Waste Regulatory Analyses (CNWRA) for the Division of Waste Management of the Nuclear Regulatory Commission (NRC), an independent agency of the United States Government. Neither the developer(s) of the code nor any of their sponsors make any warranty, expressed or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed, or represent that its use would not infringe on privately-owned right

```

drwxr-xr-x 1120/100      0 Feb 19 10:24 1998 tpa313/
-rwxr-xr-x 1120/100    1268 Jan 14 09:48 1998 tpa313/CLEANUP
-rw-r--r-- 1120/100    4190 Jun  3 14:56 1997 tpa313/README
-rw-r--r-- 1120/100   28410 Feb 19 09:26 1998 tpa313/array.f
-rw-r--r-- 1120/100   17906 Feb 19 09:26 1998 tpa313/ashplumo.f
-rw-r--r-- 1120/100   23853 Feb 19 09:26 1998 tpa313/ashrmovo.f
-rw-r--r-- 1120/100    9746 Feb 19 09:26 1998 tpa313/condxyz.f
-rw-r--r-- 1120/100   19352 Feb 19 09:26 1998 tpa313/dcags.f
-rw-r--r-- 1120/100   41420 Feb 19 09:26 1998 tpa313/dcagw.f
-rw-r--r-- 1120/100   31175 Feb 19 09:26 1998 tpa313/ebfail.f
-rw-r--r-- 1120/100   42089 Feb 19 09:26 1998 tpa313/ebrel.f
-rw-r--r-- 1120/100  212224 Feb 19 09:02 1998 tpa313/exec.f
-rw-r--r-- 1120/100    1495 Sep  3 17:54 1997 tpa313/execa.i
-rw-r--r-- 1120/100    486 Sep  3 17:54 1997 tpa313/exech.i
-rw-r--r-- 1120/100   10907 Feb 19 09:26 1998 tpa313/faulto.f
-rw-r--r-- 1120/100    5930 Feb 19 09:26 1998 tpa313/fileunit.f
-rw-r--r-- 1120/100    5825 Feb 19 09:26 1998 tpa313/findelev.f
-rw-r--r-- 1120/100   54888 Feb 19 09:26 1998 tpa313/invent.f
-rw-r--r-- 1120/100    689 Aug 26 14:06 1997 tpa313/Makefile
-rw-r--r-- 1120/100     78 Aug 16 13:32 1997 tpa313/max500yr.i
-rw-r--r-- 1120/100     99 Aug 16 13:32 1997 tpa313/maxchain.i
-rw-r--r-- 1120/100    186 Aug 16 13:32 1997 tpa313/maxnucl.i
-rw-r--r-- 1120/100    230 Sep 26 14:57 1997 tpa313/maxnsuba.i
-rw-r--r-- 1120/100    130 Aug 16 13:32 1997 tpa313/maxntime.i
-rw-r--r-- 1120/100   11658 Feb 19 09:26 1998 tpa313/mv.f
-rw-r--r-- 1120/100   52323 Feb 19 09:26 1998 tpa313/nfenv.f
-rw-r--r-- 1120/100     58 Aug 16 13:32 1997 tpa313/reflux2.i
-rw-r--r-- 1120/100     94 Aug 16 13:32 1997 tpa313/nintv.i
-rw-r--r-- 1120/100   1502 Jun 11 16:21 1997 tpa313/notice.i
-rw-r--r-- 1120/100   4810 Feb 19 09:26 1998 tpa313/numrecip.f
-rw-r--r-- 1120/100    259 Aug 16 13:34 1997 tpa313/path.i
-rw-r--r-- 1120/100   6292 Feb 19 09:26 1998 tpa313/peakfinder.f
-rw-r--r-- 1120/100   43659 Feb 19 09:26 1998 tpa313/ran.f
-rw-r--r-- 1120/100   48137 Feb 19 09:26 1998 tpa313/reader.f
-rw-r--r-- 1120/100   83817 Feb 19 09:26 1998 tpa313/sampler.f
-rw-r--r-- 1120/100   38456 Feb 19 09:26 1998 tpa313/seismo.f
-rw-r--r-- 1120/100    1708 Sep  3 17:49 1997 tpa313/setfiles.i
-rw-r--r-- 1120/100    144 Sep  3 11:09 1997 tpa313/stop.i
-rw-r--r-- 1120/100   34825 Feb 19 09:26 1998 tpa313/subarea.f
-rw-r--r-- 1120/100   67203 Feb 19 09:26 1998 tpa313/szft.f
-rw-r--r-- 1120/100     60 Aug 16 13:33 1997 tpa313/szft.i
-rw-r--r-- 1120/100   35049 Feb 19 11:41 1998 tpa313/tpa.inp
-rw-r--r-- 1120/100   44967 Sep 22 17:32 1997 tpa313/tpa.inp.meanvalues
-rw-r--r-- 1120/100   54967 Feb 19 09:26 1998 tpa313/uzflow.f
-rw-r--r-- 1120/100  106196 Feb 19 09:26 1998 tpa313/uzft.f
-rw-r--r-- 1120/100   10398 Feb 19 09:26 1998 tpa313/volcano.f
-rw-r--r-- 1120/100     60 Aug 16 13:32 1997 tpa313/ful.i
-rw-r--r-- 1120/100    430 Aug 29 18:59 1997 tpa313/fu2.i
-rw-r--r-- 1120/100     57 Aug 16 13:32 1997 tpa313/inventa.i
-rw-r--r-- 1120/100    109 Aug 16 13:32 1997 tpa313/inventb.i
-rw-r--r-- 1120/100    194 Aug 16 13:32 1997 tpa313/inventc.i
-rw-r--r-- 1120/100     62 Aug 16 13:32 1997 tpa313/inventd.i
-rw-r--r-- 1120/100     66 Aug 16 13:32 1997 tpa313/invente.i
-rw-r--r-- 1120/100     65 Aug 16 13:32 1997 tpa313/inventf.i
-rw-r--r-- 1120/100     64 Aug 16 13:32 1997 tpa313/inventg.i
-rw-r--r-- 1120/100     64 Aug 16 13:32 1997 tpa313/inventh.i
-rw-r--r-- 1120/100     75 Aug 16 13:32 1997 tpa313/inventi.i
-rw-r--r-- 1120/100     72 Aug 16 13:32 1997 tpa313/inventj.i
-rw-r--r-- 1120/100    106 Aug 16 13:32 1997 tpa313/inventk.i
-rw-r--r-- 1120/100     75 Aug 16 13:32 1997 tpa313/inventl.i
-rw-r--r-- 1120/100     78 Aug 16 13:32 1997 tpa313/inventm.i
-rw-r--r-- 1120/100    104 Aug 16 13:32 1997 tpa313/inventn.i
-rw-r--r-- 1120/100    111 Sep  4 14:39 1997 tpa313/mva.i
-rw-r--r-- 1120/100     56 Aug 16 13:32 1997 tpa313/mvb.i
-rw-r--r-- 1120/100     57 Aug 16 13:32 1997 tpa313/mvc.i

```

Files on The
TPA Version 3.1.3

Tape

DEW

2/19/98

```
-rw-r--r-- 1120/100 101 Aug 16 13:32 1997 tpa313/mvd.i
-rw-r--r-- 1120/100 72 Aug 16 13:32 1997 tpa313/mve.i
-rw-r--r-- 1120/100 72 Aug 16 13:32 1997 tpa313/mvf.i
-rw-r--r-- 1120/100 62 Aug 16 13:32 1997 tpa313/sampler0.i
-rw-r--r-- 1120/100 79 Aug 16 13:32 1997 tpa313/sampler1.i
-rw-r--r-- 1120/100 62 Aug 16 13:32 1997 tpa313/sampler2.i
-rw-r--r-- 1120/100 62 Aug 16 13:32 1997 tpa313/sampler.a.i
-rw-r--r-- 1120/100 62 Aug 16 13:32 1997 tpa313/samplerb.i
-rw-r--r-- 1120/100 62 Aug 16 13:32 1997 tpa313/samplerc.i
-rw-r--r-- 1120/100 68 Aug 16 13:32 1997 tpa313/samplerd.i
-rw-r--r-- 1120/100 133 Aug 16 13:32 1997 tpa313/sampler.e.i
-rw-r--r-- 1120/100 111 Aug 16 13:32 1997 tpa313/samplerf.i
-rw-r--r-- 1120/100 84 Aug 16 13:32 1997 tpa313/samplerg.i
-rw-r--r-- 1120/100 68 Aug 16 13:32 1997 tpa313/samplerh.i
-rw-r--r-- 1120/100 83 Aug 16 13:32 1997 tpa313/sampleri.i
drwxr-xr-x 1120/100 0 Feb 19 10:24 1998 tpa313/codes/
-rw-r--r-- 1120/100 306 Feb 17 20:25 1998 tpa313/codes/Makefile
-rw-r--r-- 1120/100 499 Jun 2 14:25 1997 tpa313/codes/README
-rw-r--r-- 1120/100 2203 Feb 17 20:25 1998 tpa313/codes/SIZES.INC
-rw-r--r-- 1120/100 164 Feb 17 20:25 1998 tpa313/codes/SIZES2.INC
-rw-r--r-- 1120/100 94534 Feb 17 20:25 1998 tpa313/codes/ashplume.f
-rw-r--r-- 1120/100 80381 Feb 17 20:25 1998 tpa313/codes/failt.f
-rw-r--r-- 1120/100 302437 Feb 18 18:55 1998 tpa313/codes/nefmks.f
-rw-r--r-- 1120/100 156101 Feb 17 20:30 1998 tpa313/codes/releaset.f
-rw-r--r-- 1120/100 171358 Feb 17 20:25 1998 tpa313/codes/snllhs.f
-rw-r--r-- 1120/100 61 Aug 16 13:32 1997 tpa313/samplerj.i
-rw-r--r-- 1120/100 208 Aug 16 13:32 1997 tpa313/samplerk.i
-rw-r--r-- 1120/100 104 Aug 16 13:32 1997 tpa313/samplerl.i
-rw-r--r-- 1120/100 63 Aug 16 13:32 1997 tpa313/samplerm.i
-rw-r--r-- 1120/100 79 Aug 16 13:32 1997 tpa313/sampler.n.i
-rw-r--r-- 1120/100 63 Aug 16 13:32 1997 tpa313/sampler.o.i
-rw-r--r-- 1120/100 120 Aug 16 13:32 1997 tpa313/samplerp.i
-rw-r--r-- 1120/100 103 Aug 16 13:32 1997 tpa313/samplerq.i
-rw-r--r-- 1120/100 176 Aug 16 13:32 1997 tpa313/sampler.r.i
-rw-r--r-- 1120/100 224 Aug 16 13:32 1997 tpa313/samplers.i
-rw-r--r-- 1120/100 70 Aug 16 13:32 1997 tpa313/sampler.t.i
-rw-r--r-- 1120/100 69 Aug 16 13:33 1997 tpa313/sampleru.i
-rw-r--r-- 1120/100 62 Aug 16 13:33 1997 tpa313/sampler.v.i
-rw-r--r-- 1120/100 60 Aug 16 13:33 1997 tpa313/sampler.w.i
-rw-r--r-- 1120/100 90 Aug 16 13:33 1997 tpa313/sampler.x.i
-rw-r--r-- 1120/100 60 Aug 16 13:33 1997 tpa313/samplery.i
-rw-r--r-- 1120/100 60 Aug 16 13:33 1997 tpa313/sampler.z.i
-rw-r--r-- 1120/100 134 Aug 16 13:33 1997 tpa313/subarea.a.i
-rw-r--r-- 1120/100 79 Aug 16 13:33 1997 tpa313/subarea.b.i
-rw-r--r-- 1120/100 82 Aug 16 13:33 1997 tpa313/subarea.c.i
-rw-r--r-- 1120/100 81 Aug 16 13:33 1997 tpa313/subarea.d.i
-rw-r--r-- 1120/100 77 Aug 16 13:33 1997 tpa313/subarea.e.i
-rw-r--r-- 1120/100 314 Aug 16 13:33 1997 tpa313/uz_climi.i
-rw-r--r-- 1120/100 926 Aug 16 13:33 1997 tpa313/uz_climr.i
-rw-r--r-- 1120/100 341 Aug 16 13:33 1997 tpa313/uz_climz.i
-rw-r--r-- 1120/100 427 Aug 16 13:33 1997 tpa313/uz_flowi.i
-rw-r--r-- 1120/100 881 Aug 16 13:33 1997 tpa313/uz_flowr.i
-rw-r--r-- 1120/100 176 Aug 16 13:33 1997 tpa313/uz_flowz.i
-rw-r--r-- 1120/100 2971 Aug 16 13:33 1997 tpa313/uz_parms.i
drwxr-xr-x 1120/100 0 Feb 19 10:24 1998 tpa313/data/
-rw-r--r-- 1120/100 1648 Jun 2 14:11 1997 tpa313/data/ebspac.nuc
-rw-r--r-- 1120/100 1274984 Aug 30 14:59 1997 tpa313/data/multiflo.dat
-rw-r--r-- 1120/100 544 Jul 3 11:28 1997 tpa313/data/seismo.dat
-rw-r--r-- 1120/100 3643 Feb 18 18:57 1998 tpa313/data/strmtube.dat
-rw-r--r-- 1120/100 850000 Aug 15 10:13 1997 tpa313/data/climato1.dat
-rw-r--r-- 1120/100 2460 Aug 15 10:13 1997 tpa313/data/climato2.dat
-rw-r--r-- 1120/100 1749 Jan 12 17:18 1998 tpa313/data/dilution.dat
-rw-r--r-- 1120/100 4586 Aug 15 10:13 1997 tpa313/data/ebsfail.def
-rw-r--r-- 1120/100 3300 Feb 10 17:34 1998 tpa313/data/ebsrel.def
-rw-r--r-- 1120/100 2260 Aug 15 10:13 1997 tpa313/data/rectedge.dat
```

```
-rw-r--r-- 1120/100 494027 Aug 15 10:13 1997 tpa313/data/soildem.dat
-rw-r--r-- 1120/100 301091 Aug 15 10:13 1997 tpa313/data/elevdem.dat
-rw-r--r-- 1120/100 1979 Feb 10 11:11 1998 tpa313/data/gs_cb_ad.dat
-rw-r--r-- 1120/100 1280 Feb 10 11:11 1998 tpa313/data/gs_cb_ci.dat
-rw-r--r-- 1120/100 1976 Feb 10 11:11 1998 tpa313/data/gs_pb_ad.dat
-rw-r--r-- 1120/100 1239 Feb 10 11:11 1998 tpa313/data/gs_pb_ci.dat
-rw-r--r-- 1120/100 2490 Feb 10 11:11 1998 tpa313/data/gw_cb_ad.dat
-rw-r--r-- 1120/100 798 Feb 10 11:11 1998 tpa313/data/gw_cb_ci.dat
-rw-r--r-- 1120/100 2491 Feb 10 11:11 1998 tpa313/data/gw_pb_ad.dat
-rw-r--r-- 1120/100 796 Feb 10 11:11 1998 tpa313/data/gw_pb_ci.dat
-rw-r--r-- 1120/100 754061 Aug 15 11:10 1997 tpa313/data/tefktd.inp
-rw-r--r-- 1120/100 85300 Feb 18 18:57 1998 tpa313/data/tpanames.dbs
drwxr-xr-x 1120/100 0 Feb 19 10:24 1998 tpa313/ccdf/
-rw-r--r-- 1120/100 22925 Jan 15 10:29 1998 tpa313/ccdf/tccdf.f
-rw-r--r-- 1120/100 66 Aug 1 14:32 1997 tpa313/ccdf/tccdf.i
-rw-r--r-- 1120/100 640 Sep 4 18:20 1997 tpa313/ccdf/tccdf.inp
-rw-r--r-- 1120/100 174 Aug 5 15:49 1997 tpa313/ccdf/Makefile
```

TPA V 3.1.4

SOFTWARE RELEASE NOTICE

| | | |
|--|----------------------|--------------------------|
| 1. SRN Number: PA-SRN-209 | | |
| 2. Project Title: TSPA & Technical Integration Code | | Project No. 20-01402-762 |
| 3. SRN Title: TPA Version 3.1.4 <i>RJ 10-4-99</i> | | |
| 4. Originator/Requestor: Ron Janetzke | | Date: Oct. 1, 1999 |
| 5. Summary of Actions <ul style="list-style-type: none"> <input type="checkbox"/> Release of new software <input type="checkbox"/> Release of modified software: <ul style="list-style-type: none"> <input type="checkbox"/> Enhancements made <input type="checkbox"/> Corrections made <input type="checkbox"/> Change of access software <input checked="" type="checkbox"/> Software Retirement | | |
| 6. Persons Authorized Access | | |
| Name | Read Only/Read-Write | Addition/Change/Delete |
| Ron Janetzke | RW | |
| Sitakanta Mohanty | RW | |
| Tim McCartin (NRC) | RW | |
| M. Rose Byrne (NRC) | RW | |
| 7. Element Manager Approval: <i>Arden Wittmayer</i> Date: <i>10/4/99</i> | | |
| 8. Remarks: Version superceded. | | |

3/17

SOFTWARE RELEASE NOTICE

| | | |
|---|-------|-----------------------------|
| 01. SRN Number: PA-SRN-169 | | |
| 02. Project Title: TSPA & Technical Integration Code | | Project No.: 20-1402-762 |
| 03. SRN Title: TPA Version 3.1.4 | | |
| 04. Originator/Requestor: Bruce Mabrito | | Date: 04/07/98 |
| 05. Summary of Actions <ul style="list-style-type: none"> <input type="checkbox"/> Release of new software <input checked="" type="checkbox"/> Release of modified software: <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Enhancements made (modest changes per NRC request) <input type="checkbox"/> Corrections made <input type="checkbox"/> Change of access software <input type="checkbox"/> Software Retirement | | |
| 06. Persons Authorized Access | | |
| Name | RO/RW | A/C/D |
| Sitakanta Mohanty | RW | |
| Ron Janetzke | RW | |
| Tim McCartin (NRC) | RW | |
| M. Rose Byrne (NRC) | RW | |
| 07. Element Manager Approval: <i>Sitakanta Mohanty for Gordon Wittmeyer</i> | | Date: 4/7/98 |
| 08. Remarks: <p>TPA Version 3.1.4 software was sent to NRC Headquarters on 04/07/98 by FTP First and also a tape is being sent via the mail.</p> | | |

SOFTWARE SUMMARY FORM

| | | | | | |
|--|--|---|---|---|--|
| 01. Summary Date: 04/07/98 | | 02. Summary prepared by (Name and phone) B. Mabrito (210)522-5149 | | 03. Summary Action: Modified | |
| 04. Software Date: 04/07/98 B. Mabrito 4/7/98 | | 05. Short Title: TPA Version 3.1.4 | | 07. Internal Software ID: None | |
| 06. Software Title: TPA - System Performance Assessment Computer Code, Version 3.1.4 | | | | | |
| 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 type="checkbox"/> Scientific/Engineering <input type="checkbox"/> Auxiliary Analyses <input checked="" type="checkbox"/> Total System PA <input type="checkbox"/> Subsystem PA <input type="checkbox"/> Other B. Specific: | |
| 11. Submitting Organization and Address: CNWRA/SWRI 6220 Culebra Road San Antonio TX 78228 | | | | 12. Technical Contact(s) and Phone: Sitakanta Mohanty (210) 522-5185 | |
| 13. Narrative: The TPA Code consists of the following modules: UZFLOW, NFENV, EBSREL, UZFT, SZFT, DCAGW, FAULTO, SEISMO, VOLCANO, ASHPLUMO, ASHRMVO, DCAGS, LHS, EXEC. | | | | | |
| 14. Computer Platform: SUN Workstation | | 15. Computer Operating System: UNIX | | 17. Number of Source Program Statements: Approx. 32,000 lines w/o Stand Alone Codes | |
| 18. Computer Memory Requirements: 45 Mb | | 19. Tape Drives: None | | 21. Graphics: N/A | |
| 20. Disk/Drum Units: N/A | | 21. Graphics: N/A | | | |
| 22. Other Operational Requirements: None | | | | | |
| 23. Software Availability: <input checked="" type="checkbox"/> Available <input type="checkbox"/> Limited <input type="checkbox"/> In-House ONLY | | | 24. Documentation Availability: <input type="checkbox"/> Available <input type="checkbox"/> Inadequate <input checked="" type="checkbox"/> In-House ONLY Including NRC | | |
| Software Custodian: <i>B. Mabrito</i> Date: <i>4/7/98</i> | | | | | |

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

13/17


DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE: TPA Version 3.1.4


April 7, 1998

Total Performance Assessment (Scientific and Engineering Software) Version 3.1.4

NOTE: This version of the TPA Software contains modest changes from the previous version released February 19, 1998. There are a total of four (4) Software Problem Change Reports (SPCRs) associated with this version change and are in the TPA Version 3.1.4 folder.

1. Scientific Notebook Documentation: Changes to the TPA code have been made utilizing SPCR's that were generated by both the CNWRA and NRC staff working on TPA Version 3.1.4.
2. Programming Language: ANSI Standard FORTRAN 77 confirmed by the Software Custodian. See attached hard copies of TPA Version 3.1.4 pages as examples.
3. Internal Documentation: On 4/7/98, B. Mabrito reviewed limited portions of the TPA Version 3.1.4 scientific and engineering software on the "Bigbend" SUN Ultra 2 work station. The TPA Version 3.1.4 software is located on the "Scratchy1" server. The portions of the TPA code reviewed contained numerous comment lines and a hard copy page showing representative comments is attached to this Design Verification Report. The internal documentation comments in TPA Version 3.1.4 meet the requirements of TOP-018, Section 5.4.4.
4. Software Labels and Data
 - a. Header Data and Format: TPA Version 3.1.4 header data and the format were compared against TOP-018 Section 5.4.6 and found acceptable.
 - b. NRC Data: TPA Version 3.1.4 NRC data and the format were compared against TOP-018, Section 5.4.6, third bullet and found to be acceptable.
 - c. Source Code Header: TPA Version 3.1.4 header data was compared to TOP-018 Section 5.4.6, fourth bullet, and found to meet the requirements and are acceptable.
5. Unique Run Identification: At the top of an output file for TPA Version 3.1.4, there is a unique identifier on the printed sheet.
6. Software Analysis and Results: The software analysis tools were not run on TPA Version 3.1.4 because the constrained nature of the changes to the code did not warrant the use of them. See the SPCR's for details of the exact changes made.

 4/7/98
Sitakanta Mohanty Date
CNWRA TPA Software Developer

 4/7/98
Bruce Mabrito Date
CNWRA Software Custodian

Attachments/

Original to: Software Folder

cc: CNWRA Software Developer/Cognizant EM

14/1/7

bigbend:/bscr3/mohanty/A_tpa3.1.4 {25} !t
tpa.e

=====

```
exec: Welcome to TPA Version 3.1.4
Job started: Tue Apr  7 11:13:17 1998
```

=====

Specified Global Parameters:

```

      NumberOfRealizations =    5
      NumberOfSubareas    =    7
      MaximumTime[yr]     = 10000.
      Volcanism scenario   =    0 (yes=1, no=0)
      Faulting scenario    =    0 (yes=1, no=0)
      Seismic scenario     =    1 (yes=1, no=0)
Distance to Critical Group (km) = 20.0
```

The specified path for data = \$TPA_DATA/
The specified path for codes = \$TPA_TEST/

To modify global parameters or the path, stop code execution using control-C

```
subarea    1 of    7           realization    1 of    5
```

15/17

```
bigbend:/bscr3/mohanty/A_tpa3.1.4 {27} vi exec.f
"exec.f" 6116 lines, 216165 characters
c Program Name:      TPA - Total-System Performance Assessment Code
c File Name:         exec.f
c File Date:         04/06/98
c Release Version:   3.1.4
c
c Client Name:       USNRC
c                   U. S. Nuclear Regulatory Commission
c                   NRC Office of Nuclear Material Safety and Safeguards
c                   Division of Waste Management
c Contract Number:   NRC 02-93-005
c
c NRC Contact        Tim McCartin (301) 415-6681
c
c CNWRA Contact:     Sitakanta Mohanty (210) 522-5185
c                   Center for Nuclear Waste Regulatory Analyses
c                   San Antonio, Texas 78238-5166
c                   smohanty@swri.edu
c
c Revisions:
c                   3.1.1 includes SPCRs 101 through 205
c                   3.1.2 includes SPCRs 206 through 224
c                   3.1.3 includes SPCRs 225 through 227
c                   3.1.4 includes SPCRs 228 through 231
c
c Documentation:     "Total-System Performance Assessment (TPA)
c                   Version 3.1 Code: Module Description and User's
c                   Guide", Center for Nuclear Waste Regulatory
c                   Analyses (in preparation)
c NUREG-Series Designator: N/A
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
```

16/17

```
c      6> TimeSeismicEvents(NumSeismicEvents)
c      7> TypeSeismicEvents(NumSeismicEvents)
c
c OUTPUT:  1> CumulatedPackageFailure(NumTime)
c
c Author:   Sui-Min Hsiung
c Date:     February 27, 1997
c Update:   April 25, 1997, May 8, 1997
c Update:   March 27 and 28, 1998
c Update:   March 29, 1998
c
c
c*****
c
cc***** sampler.i is added for temporary use *****smh, 3/29/1998****
c***** sampler0.i, samplerz.i, smaplers.i are added to allow access
c***** array samag( , ) in sampler.f &&&&& smh. 3/30/1998
c
c      include 'sampler.i'
c      include 'sampler0.i'
```

postscript

JOB 1147

Creator: Windows NT 4.0
Creation Date: 11:47 4/7/1998
Submit queue: IF 1 / Ethernet / AUTO
Submitted: Mon Apr 06 12:26:46 1998
Started: Mon Apr 06 12:26:46 1998



QMS 3825 Print System

QMS 3825 Print System

TPA V 3.2Beta

SOFTWARE RELEASE NOTICE

| | | |
|---|----------------------|--------------------------|
| 1. SRN Number: PA-SRN-210 | | |
| 2. Project Title: TSPA & Technical Integration Code | | Project No. 20-01402-762 |
| 3. SRN Title: TPA Version 3.2β Rf. 10-4-a9 | | |
| 4. Originator/Requestor: Ron Janetzke | | Date: Oct. 1, 1999 |
| 5. Summary of Actions <div style="margin-left: 20px;"> <input type="checkbox"/> Release of new software <input type="checkbox"/> Release of modified software: <div style="margin-left: 20px;"> <input type="checkbox"/> Enhancements made <input type="checkbox"/> Corrections made </div> <input type="checkbox"/> Change of access software <input checked="" type="checkbox"/> Software Retirement </div> | | |
| 6. Persons Authorized Access | | |
| Name | Read Only/Read-Write | Addition/Change/Delete |
| Ron Janetzke Sitakanta Mohanty Tim McCartin (NRC) M. Rose Byrne (NRC) | RW RW RW RW | |
| 7. Element Manager Approval: <i>Andr. Wittmeyer</i> | | Date: 10/4/99 |
| 8. Remarks: Version superceded. | | |

SOFTWARE RELEASE NOTICE

| | | |
|--|----------------------|-------------------------|
| 1. SRN Number: PA-SRN-171 | | |
| 2. Project Title: TSPA & Technical Integration Code | | Project No. 20-1402-762 |
| 3. SRN Title: TPA Version 3.2 BETA | | |
| 4. Originator/Requestor: Bruce Mabrito | | Date: 06/04/98 |
| 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 <input type="checkbox"/> Corrections made <input type="checkbox"/> Change of access software <input type="checkbox"/> Software Retirement | | |
| 6. Persons Authorized Access | | |
| Name | Read Only/Read-Write | Addition/Change/Delete |
| Sitakanta Mohanty | RW | |
| Ron Janetzke | RW | |
| Tim McCartin (NRC) | RW | |
| M. Rose Byrne (NRC) | RW | |
| 7. Element Manager Approval: <i>Gordon Wiltmayer</i> | | Date: 6/4/98 |
| 8. Remarks: TPA Version 3.2 BETA software was sent to NRC Headquarters on 06/04/98. | | |

SOFTWARE SUMMARY FORM

| | | | |
|---|---|--|--|
| 01. Summary Date: 06/04/98 | 02. Summary prepared by (Name and phone): Sitakanta Mohanty (210) 522-5185 | 03. Summary Action: Modified | |
| 04. Software Date: 04/06/98 | 05. Short Title: TPA Version 3.2 BETA | | |
| 06. Software Title: TPA - System Performance Assessment Computer Code, Version 3.2 BETA | | 07. Internal Software ID: None | |
| 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 type="checkbox"/> Scientific/Engineering <input type="checkbox"/> Auxiliary Analyses <input checked="" type="checkbox"/> Total System PA <input type="checkbox"/> Subsystem PA <input type="checkbox"/> Other b. Specific: | |
| 11. Submitting Organization and Address: CNWRA/SwRI 6220 Culebra Road San Antonio, TX 78228 | | 12. Technical Contact(s) and Phone: Sitakanta Mohanty (210) 522-5185 | |
| 13. Software Application: The TPA Code consists of the following modules: UZFLOW, NFENV, EBSREL, UZFT, SZFT, DCAGW, FAULTO, SEISMO, VOLCANO, ASHPLUMO, ASHRMVO, DCAGS, LHS, EXEC. | | | |
| 14. Computer Platform: SUN Workstation | 15. Computer Operating System: UNIX | 16. Programming Language(s): FORTRAN | 17. Number of Source Program Statements: Approx. 32,000 lines w/o Stand Alone Codes |
| 18. Computer Memory Requirements: 45 Mb | 19. Tape Drives: None | 20. Disk Units: N/A | 21. Graphics: N/A |
| 22. Other Operational Requirements: None | | | |
| 23. Software Availability: <input checked="" type="checkbox"/> Available <input type="checkbox"/> Limited <input type="checkbox"/> In-House ONLY | | 24. Documentation Availability: <input type="checkbox"/> Available <input type="checkbox"/> Preliminary <input checked="" type="checkbox"/> In-House ONLY | |
| 25. <i>Sitakanta Mohanty</i> Software Developer: | | Date: <i>6/4/98</i> | |

12/63

SOFTWARE CHANGE REPORT (SCR)

| | | |
|---|---|-------------------------|
| SCR No. (Software Developer Assigns): 232 PA-SCR-232 | Software Title and Version: TPA-System Performance Assessment Computer Code, Version 3.2 BETA <i>SEM</i> | Project No: 20-1402-762 |
| Affected Software Module(s), Description of Problem(s): Modules: exec.f, reader.f, uzft.f The TPA Code produces summary output files only for the time of simulation. It is desired that output files for a compliance period shorter than the simulation time also be generated simultaneously. This will remove the need to make two separate runs to acquire both data sets. | | |
| Change Requested by: SRD 3.2 Date: 5/8/98 | Change Authorized by (Software Developer): S. Mohanty Date: 5/8/98 | |
| Description of Implemented Change(s) (If changes not implemented, please justify): New input parameters were created to control the compliance period time steps: DurationOfCompliancePeriod NumberOfStepsInCompliancePeriod RatioOfLastToFirstTimeStepInCompliancePeriod Ten new files were created to contain the compliance period output: cumrel_c.res gscddf_c.res gwccdf_c.res gwpkdos_c.res npkdos_c.res pkreltim_c.res relccdf_c.res relgwgs_c.res totdos_c.res airpkdos_c.res | | |
| Description of Tests to Approve the Quality of the Changed Version | | |
| Implemented by: Ron Janetta | Completed Date: 5-27-98 | |

CNWRA Form TOP-5 (05/98)

NOTE: The original SCRs for TPA 3.2 B were reviewed by QA and were reviewed by The Audit Team of Audit 98-01. On 7/16/98, I provided the original SCRs to S. Mohanty so that he could update them and complete work on TPA 3.2, being shipped 7/17/98. Sundhar 7/17/98

13/63

SOFTWARE CHANGE REPORT (SCR)

An SCR must be filled up for reporting any problems found in a developed code or any necessary modifications that must be brought into the code. A revised Software Requirement Document (SRD) is needed if extensive modifications to the code are expected. If one SCR description of a group of small changes, it will be considered acceptable if specific changes are clearly identified with the modified modules.

| | | |
|---|---|-----------------------------------|
| SCR No. (Software Developer Assigns): PA-SCR# 233 | Software Title and Version: TPA 3.2 | Project No: 20-1402-762 |
| Affected Software Module(s), Description of Problem(s): The TPA code (exec.f, ashplume.f, uzst.f, ebstail.f, ebsrel.f, tpa.inp, execa.i) was modified to change the format and content of the "append option" files. | | |
| Change Requested by: TPA SRD 3.2 Date: 5/8/98 | Change Authorized by (Software Developer): S. Mohanby Date: 5/8/98 | |
| Description of Implemented Change(s) (If changes not implemented, please justify): The "append option" files (30 in total) contain all input and output values for the 13 exec.f convergence module and 4 standalone codes. The format was modified to remove repeated information and values were added that were missing. This decreased the disk space requirements (initial was ~10 meg/realization). Also, the user can select which files (only selecting 1 is allowed) or all files. The format changes were evaluated and found to be satisfactory. Attached are printouts of these files showing the new format. | | |
| Implemented by: T. Rice | Completed Date: 5/26/98 | |

14/63

SOFTWARE CHANGE REPORT (SCR)

An SCR must be filled up for reporting any problems found in a developed code or any necessary modifications that must be brought into the code. A revised Software Requirement Document (SRD) is needed if extensive modifications to the code are expected. If one SCR description of a group of small changes, it will be considered acceptable if specific changes are clearly identified with the modified modules.

| | | |
|---|--|-----------------------------------|
| SCR No. (Software Developer Assigns): PA-SCR# 234 | Software Title and Version: TPA 3.2 | Project No: 20-1402-762 |
| Affected Software Module(s), Description of Problem(s): Modify release.f to introduce multiple scenario event classes: Two separate times for faulting and volcanic events and four separate time intervals for seismic failures. Also assign separate we SF wetfractions (i.e. water heights in the WP) for each failure mode, including initially defective and, corrosion failures, and all scenario failure types. | | |
| Change Requested by: See SRD Date: 5/8/98 | Change Authorized by (Software Developer): S. Mohanty Date: 5/8/98 | |
| Description of Implemented Change(s) (If changes not implemented, please justify): New parameters were introduced for each failure type. Input file elsrnl.inp was also changed corresponding to changes in release.f. Separate SF wetfractions can be seen in the input file elsrnl.inp. | | |
| Implemented by: S. Mohanty | Completed Date: 6/4/98 | |

15/63

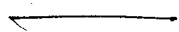
SOFTWARE CHANGE REPORT (SCR)

An SCR must be filled up for reporting any problems found in a developed code or any necessary modifications that must be brought into the code. A revised Software Requirement Document (SRD) is needed if extensive modifications to the code are expected. If one SCR description of a group of small changes, it will be considered acceptable if specific changes are clearly identified with the modified modules.

| | | |
|--|--|-----------------------------------|
| SCR No. (Software Developer Assigns): PA-SCR # 235 | Software Title and Version: TPA 3.2 | Project No: 20-1402-762 |
| Affected Software Module(s), Description of Problem(s): modify the codes (release.f, eberel.f, exec.f) to calculate ^{aqueous} gap fraction / release. | | |
| Change Requested by: See SRD for TPA 3.2 Date: 5/8/98 | Change Authorized by (Software Developer): Sitakanta Mohanty Date: 5/28/98 | |
| Description of Implemented Change(s) (If changes not implemented, please justify): release.f was modified to properly account for gap fraction release. The code used to compute it incorrectly because it was inherently assuming that all SF in the WP were wet. To demonstrate the functionality of the code, two graphs are attached. Fig. 1 has no gap fraction inventory and the 2nd fig. shows the release rate at 50% gap fraction. For other testing, see Scientific Notebook # 233. | | |
| Implemented by: S. Mohanty | Completed Date: 6/4/98 | |

16/63

SOFTWARE PROBLEM/CHANGE REPORT

| | | |
|--|--|---|
| 01. SPCR Number: PA-SCR #236 | | |
| 02. Project Title: TPA Code Testing TPA 3.2 | | 20-5708-7622 Project Number: |
| 03. SPCR Title: Allow the user to specify the aqueous nuclides | | |
| 04. AFFECTED SOFTWARE AND/OR DOCUMENTATION | | |
| SIZES.INC ebsrel.f exec.f | maxnucl.i reader.f reader.i (new lib) | releaset.f szft.f tpa.inp uzft.f |
| 05. DESCRIPTION OF PROBLEM/CHANGE | | |
| The "tpa.inp" aqueous nuclides in v. 3.1.4 were fixed and no other nuclides and chains could be analyzed. | | |
| 06. PROBLEM SOLUTION/NEED FOR CHANGE | | |
| By allowing the user to select which chains and nuclides to analyze in a tpa run, the user can focus on specific nuclides and chains — thus saving time in generating results. | | |
| 07. Originator: SRD 3.2 | Title:  | Date: 5/8/98 |
| PROJECT | | |
| 08. Need by Date: 6/4/98 | 09. Approved: S. Mohanty | Disapproved: |
| 10. Software Developer: S. Mohanty et al. | Date: 5/8/98 | |
| 11. Element Manager: | Date: 6/4/98 | |
| 12. IMPLEMENTED SOLUTION | | |
| the tpa code was modified to allow the user to select any nuclides and chains in INVENT. | | |
| 13. Implemented By: R. Ricci | | Date: 5/19/98 |

17/63

SOFTWARE CHANGE REPORT (SCR)

An SCR must be filled up for reporting any problems found in a developed code or any necessary modifications that must be brought into the code. A revised Software Requirement Document (SRD) is needed if extensive modifications to the code are expected. If one SCR description of a group of small changes, it will be considered acceptable if specific changes are clearly identified with the modified modules.

| | | |
|---|---|----------------------------|
| SCR No. (Software Developer Assigns): PA-SCR # 237 | Software Title and Version: TPA 3.2 | Project No: 20-1402-762 |
| Affected Software Module(s), Description of Problem(s): The "exec.f" and "tpa.inp" files were modified to perform only direct release calculations only. | | |
| Change Requested by: Post SRD addition from NRC Date: 5/18/98 | Change Authorized by (Software Developer): S. Mohanly Date: 5/18/98 | |
| Description of Implemented Change(s) (If changes not implemented, please justify): "exec.f" and "tpa.inp" were modified to add a flag to allow the user to select direct release calculations only. This flag will allow the user to obtain ground surface (direct release) results much quicker than previously since ground water-related computations are not performed. The check conducted to verify the correct implementation of these changes is provided in Attachments 1 and 2 which included screen print and peak air doses for all calculations and for direct release calculations only. | | |
| Implemented by: R. Ricci | Completed Date: 5/25/98 | |

18/63

SOFTWARE CHANGE REPORT (SCR)

An SCR must be filled up for reporting any problems found in a developed code or any necessary modifications that must be brought into the code. A revised Software Requirement Document (SRD) is needed if extensive modifications to the code are expected. If one SCR description of a group of small changes, it will be considered acceptable if specific changes are clearly identified with the modified modules.

| | | |
|--|---|-----------------------------------|
| SCR No. (Software Developer Assigns): PA-SCR #238 | Software Title and Version: TPA 3.2 | Project No: 20-1402-762 |
| Affected Software Module(s), Description of Problem(s): "ebfail.f", "tpa.inp" and "exec.f" were modified to allow the user to specify a zero initial failure time. | | |
| Change Requested by: see SRD for TPA 3.2 Date: 5/8/98 | Change Authorized by (Software Developer): S. Mohanty Date: 5/4/98 | |
| Description of Implemented Change(s) (If changes not implemented, please justify): The TPA input file (tpa.inp), and ^{ebfail.f} exec.f were modified to allow for user-selected initial failure times, instead of always having the initial failure time at 0 yr. This change allows the user to investigate mechanisms for wp failure at times other than 0 yr and the impacts on TPA results. Testing demonstrates the correct implementation of this change for $t=0$ yr initial failure time (Attachment 1) and $t=4000$ yr (Attachment 2) in the | | |
| Implemented by: R. Rice | Completed Date: 5/21/98 screen print, the input param value, and the release input file (ebfail.f) | |

19/63

SOFTWARE CHANGE REPORT (SCR)

An SCR must be filled up for reporting any problems found in a developed code or any necessary modifications that must be brought into the code. A revised Software Requirement Document (SRD) is needed if extensive modifications to the code are expected. If one SCR description of a group of small changes, it will be considered acceptable if specific changes are clearly identified with the modified modules.

| | | |
|---|--|-----------------------------------|
| SCR No. (Software Developer Assigns): PA-SCR #239 | Software Title and Version: TPA 3.2 | Project No: 20-1402-762 |
| Affected Software Module(s), Description of Problem(s): Diffusion calculations in release-f takes about 3 second expiration ^{cpu time} in a 50,000 yr run. For a 200 vector 50,000 yr, 7 subarea run it takes 3 hours of additional time. Because diffusion does not make a strong contribution to the release rate, therefore disable it in release-f and get rid of corresponding parameters in tpa.inpad and rel. | | |
| Change Requested by: R. Cadell of NRC Date: 5/24/98 | Change Authorized by (Software Developer): S. Mohanty Date: 5/24/98 | |
| Description of Implemented Change(s) (If changes not implemented, please justify): Diffusion calculations were disabled by disabling diffux subroutine. Note that for a given set of specified parameters, not release rate could be negative when diffusion is ON. Tests were conducted and we concluded that when the concentration in the backfill/surrounding rock is more than that in the wt, radionuclide back-diffuse back to the container resulting in -ve values. | | |
| Implemented by: S. Mohanty | Completed Date: 5/31/98 | |


CNWRA TOP-5 Form

-ve values
see appendix
figure
Also in
J. Bogen's
scientific
notebook #
233.

20/63

SOFTWARE CHANGE REPORT (SCR)


An SCR must be filled up for reporting any problems found in a developed code or any necessary modifications that must be brought into the code. A revised Software Requirement Document (SRD) is needed if extensive modifications to the code are expected. If one SCR description of a group of small changes, it will be considered acceptable if specific changes are clearly identified with the modified modules.

| | | |
|--|---|-----------------------------------|
| SCR No. (Software Developer Assigns): PA-SCR #240 | Software Title and Version: TPA 3.2 | Project No: 20 1402 762 |
| Affected Software Module(s), Description of Problem(s): ebsrel.f, ebsrel.def / ebsrel.inp Need new parameters for height fraction of wet SF in scenario defective WPs. Need finer resolution of seismic failure times for scenario events write these to ebsrel.inp | | |
| Change Requested by: See TPA 3.2 SRD Date: 5/8/98 | Change Authorized by (Software Developer): S. Mohanty Date: 6/4/98 5/8/98 | |
| Description of Implemented Change(s) (If changes not implemented, please justify): Added new input parameters for each event (SFWettedFraction <event>) modified ebsrel.f to read in new params and get failure times passed in so that these new data can be written to ebsrel.inp | | |
| Implemented by:  | Completed Date: May 25, 1998 | |

21/63

SOFTWARE CHANGE REPORT (SCR)

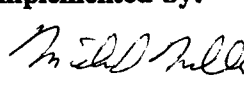
An SCR must be filled up for reporting any problems found in a developed code or any necessary modifications that must be brought into the code. A revised Software Requirement Document (SRD) is needed if extensive modifications to the code are expected. If one SCR description of a group of small changes, it will be considered acceptable if specific changes are clearly identified with the modified modules.

| | | |
|---|---|--------------------|
| SCR No. (Software Developer Assigns): PA-SCR #241 | Software Title and Version: | Project No: |
| Affected Software Module(s), Description of Problem(s): ebsrel.f, ebsfilt.f, ebsfilt.def / ebsfilt.inp Add ebsfilt as a standalone to tpa code | | |
| Change Requested by: See TPA3.2 SRD Date: 5/8/98 | Change Authorized by (Software Developer): S. Mohan Date: 6/4/98 5/8/98 | |
| Description of Implemented Change(s) (If changes not implemented, please justify): Add ebsfilt to ebsrel.f. Add new parameters. Also write out other values to ebfilt input file (ebsfilt.inp) Ebsfilt writes out ebsnef2.dat so ebsrel now reads that in instead of ebsnef.dat. | | |
| Implemented by:  | Completed Date: May 27 1998 | |

22/63

SOFTWARE CHANGE REPORT (SCR)

An SCR must be filled up for reporting any problems found in a developed code or any necessary modifications that must be brought into the code. A revised Software Requirement Document (SRD) is needed if extensive modifications to the code are expected. If one SCR description of a group of small changes, it will be considered acceptable if specific changes are clearly identified with the modified modules.

| | | |
|---|--|-----------------------------------|
| SCR No. (Software Developer Assigns): PA-SCR #242 | Software Title and Version: TPA 3.2 | Project No: 20 1402 762 |
| Affected Software Module(s), Description of Problem(s): nfev.f, reflux3.f, dry-thick.dat Add reflux3 to be used as a selectable model (can now choose 1, 2, or 3) | | |
| Change Requested by: See TPA 3.2 SRD Date: 5/8/98 | Change Authorized by (Software Developer): S. Mohan Date: 6/4/98 5/8/98 | |
| Description of Implemented Change(s) (If changes not implemented, please justify): Reflux3 uses some reflux2 params but also needs some new ones added: WPUntCell Width, Fraction of Condensate... reflux3 was highly modified to fit into nfev because of several differences in its standalone implementation and its use as a subroutine. Most changes center on the input variables and variables passed between reflux3 subroutines. All REAL's were changed to double precision. No mods were made to computational parts. | | |
| Implemented by:  | Completed Date: June 4, 1998 | |

23/63

SOFTWARE CHANGE REPORT (SCR)

An SCR must be filled up for reporting any problems found in a developed code or any necessary modifications that must be brought into the code. A revised Software Requirement Document (SRD) is needed if extensive modifications to the code are expected. If one SCR description of a group of small changes, it will be considered acceptable if specific changes are clearly identified with the modified modules.

| | | |
|--|--|-----------------------------------|
| SCR No. (Software Developer Assigns): PA-SCR # 243 | Software Title and Version: TPA 3.2 | Project No: 20-1402-762 |
| Affected Software Module(s), Description of Problem(s): Modify release.f, to ebsrel.f, Exec and tpa.inp so that the user has the option to specify additional non-zero time for initially defective failure. | | |
| Change Requested by: Post-SRD instruction from the NRC Date: 8/16/98 | Change Authorized by (Software Developer): S. Mohanty Date: 8/16/98 | |
| Description of Implemented Change(s) (If changes not implemented, please justify): release.f was modified to read initially defective failure time from the input file ebsrel.inp. The change was tested within the TPA code. | | |
| Implemented by: S. Mohanty | Completed Date: 6/4/98 | |

24/63

SOFTWARE CHANGE REPORT (SCR)

An SCR must be filled up for reporting any problems found in a developed code or any necessary modifications that must be brought into the code. A revised Software Requirement Document (SRD) is needed if extensive modifications to the code are expected. If one SCR description of a group of small changes, it will be considered acceptable if specific changes are clearly identified with the modified modules.

| | | |
|---|---|-----------------------------------|
| SCR No. (Software Developer Assigns): PA-SCR-2474 6/7/98 | Software Title and Version: TPA 3-2 | Project No: 20-1402-762 |
| Affected Software Module(s), Description of Problem(s): Invert as a barrier to release. Add a module to EBSREL.f so that advective flow and transport through an invert can be computed. | | |
| Change Requested by: See SRD Date: 5/5/98 | Change Authorized by (Software Developer): S. Mohanty Date: 6/4/98 | |
| Description of Implemented Change(s) (If changes not implemented, please justify): ebsfilt.f module was developed by R. Codell to incorporate a convolution integral calculation for transport through the invert. ebsfilt.f is left as a stand-alone code in/code and is invoked in ebsrel.f after release of | | |
| Implemented by: R. Codell (ebsfilt.f) S. Mohanty et al. (ebsrel.f) | Completed Date: 6/4/98 | |

25/63

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE: TPA Version 3.2BETA

June 4, 1998

Total-System Performance Assessment (Scientific and Engineering Software) Version 3.2BETA

NOTE: This BETA version of the TPA Software contains changes from the previous version released April 6, 1998. Software Change Reports (SCRs) have been utilized as the change documentation method and they are being retained in the TPA Version 3.2 BETA folder.

1. This Design Verification Report is prepared by: Bruce Mabrito in conjunction with the Software Development Team.

Full Title of CNWRA scientific and engineering software: Total-System Performance Assessment (TPA) Version 3.2BETA.

Demonstration work station: SPARC20 in conjunction with the SCRATCHY1 server.

Operating System: SunOS 5.5.1

2. Software Requirements Description and any changes thereto approved by Element Manager?

☒ YES

☐ NO

☐ N/A

If no, explain:

3. Software Development Plan (SDP) and any changes have been approved by the Element Manager?

☒ YES

☐ NO

☐ N/A

If no, explain:

4. Design and Development

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

☒ YES

☐ NO

☐ N/A

If no, explain:

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

☒ YES

☐ NO

☐ N/A

If no, explain:

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

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 labelling 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

If no, explain: Per NRC instructions/schedule and the CNWRA HLW Operations Plans, the TPA Version 3.2 Users' Manual will be delivered to the NRC later.

Are there basic instructions for the use of the software?

☐ YES

☒ NO

☐ N/A

If no, explain: TPA Version 3.2BETA follows the Draft Users' Manual developed for TPA Version 3.1.4 and that Users' Manual will be delivered to the NRC in June, 1998.

10. Acceptance Testing

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

☐ YES

☐ NO

☒ N/A

If no, explain: NRC will perform the testing because this is a BETA version.

27/63

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

YES

NO

N/A

If no, explain: This acceptance testing will not be accomplished for TPA Version 3.2BETA.

Have installation tests been performed on the target platform?

YES

NO

N/A

If no, explain: See the CNWRA TPA Version 3.2 SDP. Installation testing was performed on the BigBend server.

11. Configuration Control

Is the Software Summary Form completed and signed?

YES

NO

N/A

If no, explain:

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

YES

NO

N/A

If no, explain: The technical description is given in the Users' Manual for TPA Version 3.1.4.

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

YES

NO


N/A


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

YES

NO

N/A


Sitakanta Mohanty Date
CNWRA TPA Software Developer


Bruce Mabrito Date
CNWRA Software Custodian

Attachments/

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

```
=====
exec: Welcome to TPA Version 3.2BETA
Job started: Thu Jun 4 16:46:22 1998
=====
```

DATE, TIME, NAME,
Output File VERSION

Specified Global Parameters:

```
Compliance Period = 10000.0 (yr)
Maximum Simulation Time = 10000.0 (yr)
Number Of Realizations = 1
Number Of Subareas = 7
Volcanism scenario = 1 (yes=1, no=0)
Faulting scenario = 1 (yes=1, no=0)
Seismic scenario = 1 (yes=1, no=0)
Distance to Critical Group = 20.0 (km)
```

```
**>>> CAUTION: CHECKING OF NUCLIDES AND CHAINS IS DISABLED <<<**
**>>> You may not be using the standard chains specified <<<**
**>>> in the invent module. <<<**
**>>> (see "CheckNuclidesAndChains(yes=1,no=0)" in tpa.inp)<<<**
```

```
***>>> NOTE: When running with volcanism, verify that <<<***
***>>> the maximum value of the PDF for parameter <<<***
***>>> TimeOfNextVolcanicEventinRegionOfInterest[yr] is <<<***
***>>> equal to the parameter MaximumTime[yr]. <<<***
```

```
The specified path for data = $TPA_DATA/
The specified path for codes = $TPA_TEST/
```

```
**To modify global parameters or the path, stop code execution using control-C**
```

29/63

bigbend:/bscr3/mohanty/A_tpa3.2beta {394} !t
tpa.e

=====

exec: Welcome to TPA Version 3.2BETA
Job started: Thu Jun 4 18:34:13 1998

=====

Specified Global Parameters:

Compliance Period = 10000.0 (yr)
Maximum Simulation Time = 10000.0 (yr)
Number Of Realizations = 1
Number Of Subareas = 7
Volcanism scenario = 0 (yes=1, no=0)
Faulting scenario = 0 (yes=1, no=0)
Seismic scenario = 1 (yes=1, no=0)
Distance to Critical Group = 20.0 (km)

>>> CAUTION: CHECKING OF NUCLIDES AND CHAINS IS DISABLED <<<
>>> You may not be using the standard chains specified <<<
>>> in the invent module. <<<
>>> (see "CheckNuclidesAndChains(yes=1,no=0)" in tpa.inp)<<<

The specified path for data = \$TPA_DATA/
The specified path for codes = \$TPA_TEST/

To modify global parameters or the path, stop code execution using control-C

subarea 1 of 7 realization 1 of 1

exec: calling uzflow
exec: calling nfenv
exec: calling ebsfail

*Installed on a
Platform similar to
that of the NRC.
Gru
6/4/98*

```
c Program Name:      TPA - Total-System Performance Assessment Code
c File Name:         %M%
c File Date:         %G%
c Release Version:   3.2BETA
c
c Client Name:       USNRC
c                   U. S. Nuclear Regulatory Commission
c                   NRC Office of Nuclear Material Safety and Safeguards
c                   Division of Waste Management
c Contract Number:   NRC 02-93-005
c
c NRC Contact        Tim McCartin (301) 415-6681
c
c CNWRA Contact:     Sitakanta Mohanty (210) 522-5185
c                   Center for Nuclear Waste Regulatory Analyses
c                   San Antonio, Texas 78238-5166
c                   smohanty@swri.edu
c
c Revisions:
c                   3.1.1 includes SPCRs 101 through 205
c                   3.1.2 includes SPCRs 206 through 224
c                   3.1.3 includes SPCRs 225 through 227
c                   3.1.4 includes SPCRs 228 through 231
c                   3.2BETA includes SPCRs 232 through 243
c
c Documentation:     "Total-System Performance Assessment (TPA)
c                   Version 3.1 Code: Module Description and User's
c                   Guide", Center for Nuclear Waste Regulatory
c                   Analyses (in preparation)
c NUREG-Series Designator: N/A
c cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c = = = = =
c
c                   D I S C L A I M E R
c
c = = = = =
c
c "This computer code/material was prepared 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. Neither the developer(s) of the code nor any of their
c sponsors 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 privately-
c owned rights."
c
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 the program (including but not limited to loss of
c data or data being rendered inaccurate or losses sustained by third
c parties or a failure of the program to operate with other programs),
c even if you have been advised of the possibility of such damages or
c for any claim by any other party."
c
c = = = = =
c
c by S. Mohanty, R. Janetzke, R. Rice, A. Lozano
c R. Manteufel (initial version)
c
c=====
```

program exec

```
c=====
c Executive for TPA Version 3.2BETA
c Contact Person: : S. Mohanty
c
cc      1      2      3      4      5      6      7
cc3456789012345678901234567890123456789012345678901234567890
```

temp

a tpa32beta/ 0 tape blocks
a tpa32beta/CLEANUP 3 tape blocks
a tpa32beta/array.f 56 tape blocks
a tpa32beta/ashplumo.f 40 tape blocks
a tpa32beta/ashrmovo.f 47 tape blocks
a tpa32beta/condxyz.f 20 tape blocks
a tpa32beta/dcags.f 42 tape blocks
a tpa32beta/dcagw.f 85 tape blocks
a tpa32beta/ebsfail.f 67 tape blocks
a tpa32beta/ebsrel.f 119 tape blocks
a tpa32beta/exec.f 602 tape blocks
a tpa32beta/execa.i 4 tape blocks
a tpa32beta/execb.i 1 tape blocks
a tpa32beta/faulto.f 22 tape blocks
a tpa32beta/fileunit.f 12 tape blocks
a tpa32beta/findelev.f 12 tape blocks
a tpa32beta/invent.f 108 tape blocks
a tpa32beta/Makefile 2 tape blocks
a tpa32beta/max500yr.i 1 tape blocks
a tpa32beta/maxchain.i 1 tape blocks
a tpa32beta/maxnnucl.i 1 tape blocks
a tpa32beta/maxnsuba.i 1 tape blocks
a tpa32beta/maxntime.i 1 tape blocks
a tpa32beta/mv.f 23 tape blocks
a tpa32beta/nfenv.f 143 tape blocks
a tpa32beta/reflux2.i 1 tape blocks
a tpa32beta/nintv.i 1 tape blocks
a tpa32beta/notice.i 3 tape blocks
a tpa32beta/numrecip.f 10 tape blocks
a tpa32beta/path.i 1 tape blocks
a tpa32beta/peakfinder.f 13 tape blocks
a tpa32beta/ran.f 86 tape blocks
a tpa32beta/reader.f 152 tape blocks
a tpa32beta/reader.i 1 tape blocks
a tpa32beta/sampler.f 165 tape blocks
a tpa32beta/seismo.f 108 tape blocks
a tpa32beta/setfiles.i 4 tape blocks
a tpa32beta/stop.i 1 tape blocks
a tpa32beta/subarea.f 69 tape blocks
a tpa32beta/szft.f 134 tape blocks
a tpa32beta/szft.i 1 tape blocks
a tpa32beta/tpa.inp 86 tape blocks
a tpa32beta/tpa.inp.meanvalues 88 tape blocks
a tpa32beta/uzflow.f 108 tape blocks
a tpa32beta/uzft.f 219 tape blocks
a tpa32beta/volcano.f 21 tape blocks
a tpa32beta/ful.i 1 tape blocks
a tpa32beta/fu2.i 1 tape blocks
a tpa32beta/inventa.i 1 tape blocks
a tpa32beta/inventb.i 1 tape blocks
a tpa32beta/inventc.i 1 tape blocks
a tpa32beta/inventd.i 1 tape blocks
a tpa32beta/invente.i 1 tape blocks
a tpa32beta/mva.i 1 tape blocks
a tpa32beta/inventf.i 1 tape blocks
a tpa32beta/inventg.i 1 tape blocks
a tpa32beta/inventh.i 1 tape blocks
a tpa32beta/inventi.i 1 tape blocks
a tpa32beta/inventj.i 1 tape blocks
a tpa32beta/inventk.i 1 tape blocks
a tpa32beta/inventl.i 1 tape blocks
a tpa32beta/inventm.i 1 tape blocks
a tpa32beta/inventn.i 1 tape blocks
a tpa32beta/mvb.i 1 tape blocks
a tpa32beta/mvc.i 1 tape blocks

MAKE FILE
REFERENCE

a tpa32beta/mvd.i 1 tape blocks
a tpa32beta/mve.i 1 tape blocks
a tpa32beta/mvf.i 1 tape blocks
a tpa32beta/sampler0.i 1 tape blocks
a tpa32beta/sampler1.i 1 tape blocks
a tpa32beta/sampler2.i 1 tape blocks
a tpa32beta/sampler3.i 1 tape blocks
a tpa32beta/sampler.a.i 1 tape blocks
a tpa32beta/samplerb.i 1 tape blocks
a tpa32beta/samplerc.i 1 tape blocks
a tpa32beta/samplerd.i 1 tape blocks
a tpa32beta/sampler.e.i 1 tape blocks
a tpa32beta/samplerf.i 1 tape blocks
a tpa32beta/samplerg.i 1 tape blocks
a tpa32beta/samplerh.i 1 tape blocks
a tpa32beta/sampleri.i 1 tape blocks
a tpa32beta/samplerj.i 1 tape blocks
a tpa32beta/samplerk.i 1 tape blocks
a tpa32beta/samplerl.i 1 tape blocks
a tpa32beta/samplerm.i 1 tape blocks
a tpa32beta/samlern.i 1 tape blocks
a tpa32beta/sampler.o.i 1 tape blocks
a tpa32beta/samplerp.i 1 tape blocks
a tpa32beta/samplerq.i 1 tape blocks
a tpa32beta/samlerr.i 1 tape blocks
a tpa32beta/samplers.i 1 tape blocks
a tpa32beta/samlert.i 1 tape blocks
a tpa32beta/sampleru.i 1 tape blocks
a tpa32beta/samlerv.i 1 tape blocks
a tpa32beta/samplerw.i 1 tape blocks
a tpa32beta/samplerx.i 1 tape blocks
a tpa32beta/samlery.i 1 tape blocks
a tpa32beta/samplerz.i 1 tape blocks
a tpa32beta/subareaa.i 1 tape blocks
a tpa32beta/subareab.i 1 tape blocks
a tpa32beta/subareac.i 1 tape blocks
a tpa32beta/subaread.i 1 tape blocks
a tpa32beta/subareae.i 1 tape blocks
a tpa32beta/uz_climi.i 1 tape blocks
a tpa32beta/uz_climr.i 2 tape blocks
a tpa32beta/uz_climz.i 1 tape blocks
a tpa32beta/uz_flowi.i 1 tape blocks
a tpa32beta/uz_flowr.i 2 tape blocks
a tpa32beta/uz_flowz.i 1 tape blocks
a tpa32beta/uz_parms.i 6 tape blocks
a tpa32beta/codes/ 0 tape blocks
a tpa32beta/codes/Makefile 1 tape blocks
a tpa32beta/codes/README 1 tape blocks
a tpa32beta/codes/SIZES.INC 5 tape blocks
a tpa32beta/codes/SIZES2.INC 1 tape blocks
a tpa32beta/codes/ashplume.f 185 tape blocks
a tpa32beta/codes/failt.f 157 tape blocks
a tpa32beta/codes/nefmks.f 591 tape blocks
a tpa32beta/codes/releaset.f 333 tape blocks
a tpa32beta/codes/snllhs.f 335 tape blocks
a tpa32beta/codes/ebsfilt.f 7 tape blocks
a tpa32beta/data/ 0 tape blocks
a tpa32beta/data/multiflo.dat 2491 tape blocks
a tpa32beta/data/strmtube.dat 9 tape blocks
a tpa32beta/data/climatol.dat 1661 tape blocks
a tpa32beta/data/climato2.dat 5 tape blocks
a tpa32beta/data/dilution.dat 4 tape blocks
a tpa32beta/data/ebsfail.def 9 tape blocks
a tpa32beta/data/ebsrel.def 9 tape blocks
a tpa32beta/data/rectedge.dat 5 tape blocks

a tpa32beta/data/soildem.dat 965 tape blocks
a tpa32beta/data/elevdem.dat 589 tape blocks
a tpa32beta/data/gs_cb_ad.dat 6 tape blocks
a tpa32beta/data/gs_cb_ci.dat 5 tape blocks
a tpa32beta/data/gs_pb_ad.dat 6 tape blocks
a tpa32beta/data/gs_pb_ci.dat 5 tape blocks
a tpa32beta/data/gw_cb_ad.dat 7 tape blocks
a tpa32beta/data/gw_cb_ci.dat 3 tape blocks
a tpa32beta/data/gw_pb_ad.dat 7 tape blocks
a tpa32beta/data/gw_pb_ci.dat 3 tape blocks
a tpa32beta/data/tefkfi.inp 1473 tape blocks
a tpa32beta/data/tpanames.dbs 182 tape blocks
a tpa32beta/data/ebsfilt.def 2 tape blocks
a tpa32beta/data/dry_thick.dat 1 tape blocks
a tpa32beta/ccdf/ 0 tape blocks
a tpa32beta/ccdf/tccdf.f 45 tape blocks
a tpa32beta/ccdf/tccdf.i 1 tape blocks
a tpa32beta/ccdf/tccdf.inp 2 tape blocks
a tpa32beta/ccdf/Makefile 1 tape blocks

SOFTWARE DEVELOPMENT PLAN

45/63

TPA Version
3.2
Folder

**SOFTWARE DEVELOPMENT PLAN
FOR TOTAL-SYSTEM PERFORMANCE ASSESSMENT VERSION 3.2 CODE
May 29, 1998**

This software development plan (SDP) describes the approach to be followed in implementing the code modifications to be made to the Total-system Performance Assessment (TPA) Version 3.1.4 code to meet the design specifications for the TPA Version 3.2 code that are outlined in the software requirements description (SRD) (appendix A).

1.0 SCOPE

The scope of this software development project is described in detail in the appendix A. Two levels of software function changes will be made: (i) system-level changes that improve the flexibility of the TPA Version 3.2 code and (ii) changes to the conceptual models that underlie the consequence modules. System-level changes to be made will allow the code to produce summarized output at two time periods in a single run, to permit the user to specify a reduced number of groundwater radionuclides, and to provide the option to compute doses by the air pathway only. The conceptual model changes include: (i) modifications to the SEISMO module to produce more reasonable estimates of waste package failures due to seismic events, (ii) improvements to the EBSREL module to account for the effects of secondary minerals in the spent fuel and the presence of the concrete invert on radionuclide releases, (iii) incorporation of a modified thermal reflux model, (iv) modifications to the probability distributions for radionuclide distribution coefficients to account for correlation, and (v) improvements to the dose conversion factors (DCF) used in the DCAGW module. It should be noted that code testing is a joint activity between NRC and the CNWRA. Some of the activities are the responsibility of the NRC staff.

2.0 BASELINE ITEMS

The products to be delivered from this software development project include: (i) a beta test version of the TPA Version 3.2 code to be delivered to NRC on 6/5/98, (ii) a tested TPA Version 3.2 code to be delivered to NRC on 7/17/98, (iii) an updated version of the input file *tpa.inp* that includes the modified probability distribution functions and correlations for the radionuclide distribution coefficients, and (vi) corrected DCFs in input files *gs_cb_ad.dat*, *gs_cb_ca.dat*, *gs_pb_ad.dat*, *gs_pg_ci.dat*, *gw_cb_ad.dat*, *gw_cb_ca.dat*, *gw_pb_ci.dat*, and *bw_pb_ci.dat*. Expectations for the levels of testing to have been met for each delivery date have been specified by NRC in the letter included as appendix B.

3.0 PROJECT MANAGEMENT

Software development project tasks, schedules, staff, and provisions for reducing associated risk are discussed in this section.

3.1 Work Breakdown Structure

In Task 1 changes will be made to the TPA Version 3.1.4 code that meet the specifications for the TPA Version 3.2 code that are outlined in the SRD (appendix A). This task will be completed by June 5, 1998 and will meet the deliverable expectations stated by NRC in appendix B, namely that "[t]he June deliverable is acceptable if the code can be compiled on NRC's computing system, executed with the supplied input file, and successfully produce the *.res files." In addition NRC has specified that "[c]orrectness of the

*.res files is desired but not required for acceptability of the Beta version.” Specific Task 1 activities are

- Modification to the *tpa.inp* file to include new probability distributions and correlation coefficients for the uranium, neptunium, and plutonium distribution coefficients
- Prepare and test FORTRAN code that incorporates changes to the conceptual model and sampled parameters in the SEISMO module
- Prepare and test FORTRAN code that incorporates changes to the conceptual model and sampled parameters in the EBSREL module (NRC staff responsibility)
- Prepare and test FORTRAN code that allows the user to produce summary results at two time periods
- Prepare and test FORTRAN code that allows the user to specify the suite of groundwater radionuclides to be tracked during a simulation
- Prepare and test FORTRAN code that allows the user to specify calculation of dose by the air pathway only
- Prepare and test FORTRAN code that incorporates new conceptual model for thermally-driven moisture redistribution
- Prepare and test FORTRAN code that allows the user to specify the failure time for initially defective waste packages
- Prepare and test FORTRAN code that defines four distinct release periods for seismically-failed waste packages and two different release periods for faulting and volcanic events
- Ensure that the TPA Beta Version 3.2 code can be compiled and linked on the NRC computerized risk assessment and data analysis lab (CRADAL) system (may require assistance from NRC staff)
- Support NRC in ensuring that the TPA Beta Version 3.2 code can be executed with the modified input files on the NRC CRADAL computer system¹
- Support NRC in ensuring that the TPA Beta Version 3.2 code can successfully produce all *.res files on the NRC CRADAL computer system¹

¹ It is assumed that successful compilation, linking, and execution of TPA Version 3.2 on a Sun UltraSPARC 2 is a sufficient demonstration of the code's ability to be used on the NRC CRADAL system. However, if needed, a CNWRA staff member will travel to NRC to install the code on the CRADAL system.

In Task 2, the TPA Version 3.2 code will be tested to ensure the correctness of the *.res files. This task will be completed by July 17, 1998 and will meet the deliverable expectations stated by NRC in appendix B, namely that "[t]he July deliverable is acceptable if the code can be compiled on NRC's computing system, executed with the supplied input file, and the *.res files are correct." Moreover, as specified in appendix B, "[a]ll corrections identified from [the] joint testing effort should also be implemented." Specific Task 2 activities are

- Define suite of tests to be conducted to rigorously evaluate whether code modifications are producing anticipated intermediate outputs (input required from NRC staff)
- Conduct the defined suite of tests and examine intermediate and final outputs from single realization runs (input required from NRC staff)
- Modify FORTRAN code to produce correct intermediate and final outputs
- Ensure that the final TPA Version 3.2 code can be compiled and linked on the NRC CRADAL computer system (may require assistance from NRC staff)
- Ensure that the final TPA Version 3.2 code can be executed with the modified input files on the NRC CRADAL computer system
- Ensure that all *.res files produced by the final TPA Version 3.2 code on the NRC CRADAL computer system are correct

3.2 Schedules

The following schedules will or have been adopted in order to meet the Task 1 and 2 deliverables.

3.2.1 Task 1

- Code changes to consequence modules with verification testing completed by 6/1/98 (requires NRC staff assistance)
- Modifications to *tpa.inp* and DCF files by 6/2/98
- System-level code modifications with verification testing completed by 6/3/98
- Electronic transfer of the TPA Beta Version 3.2 code to NRC's CRADAL computer system with test compilation and execution by 6/3/98 (requires NRC staff assistance)
- Preparation of tar tape and completion of applicable QA forms by 6/4/98

3.2.2 Task 2

- Definition of test simulations provided by NRC to CNWRA by 6/8/98 (requires NRC staff assistance)

- Conduct initial test simulations 6/22/98 (requires NRC staff participation)
- Implement changes to code and input files 7/3/98
- Verification testing of code 7/10/98
- Electronic transfer of the TPA ~~Beta~~ Version 3.2 code to NRC's CRADAL system with test compilation and execution by 7/14/98 (may require NRC staff assistance)
- Preparation of tape archive and completion of applicable QA forms by 7/16/98

3.3 Staffing

In addition to PA staff who have been involved in the development of the TPA Version 3.1.4 code, completion of this software development project will require the use of consultants and SwRI personnel who are proficient in developing, implementing, and testing FORTRAN 77 code. For Task 1, key CNWRA and NRC technical staff members from appropriate KTIs will be required to produce the algorithms and data needed for modifications to the consequence modules. These KTI staff members will also participate in the final code verification testing to be completed under Task 2.

3.4 Risk Management

The primary risk associated with this project is failure to meet the June 5, 1998 deliverable date for Task 1. Those code additions and modifications that cannot be made to compile, link, and execute on CRADAL prior to June 5, 1998 will not be implemented until the testing in Task 2 is underway. NRC will be informed at the time of delivery that a certain feature was not implemented to ensure that a working code could be delivered. Because NRC does not expect a fully functional code for Task 1, the risk associated with functionality of the TPA Beta Version 3.2 code is non-zero but small.

4.0 DEVELOPMENT PROCEDURES

This section describes plans for developing the TPA Beta Version 3.2 code.

4.1 Hardware and Software Resources

All code development will be done on Sun SPARC 10 and 20, and Sun UltraSPARC 1 and 2 workstations running SunOS 5.3.1. The Sun FORTRAN 77 Version 4.2 compiler will be used.

4.2 Software Development Lifecycle

The project will consist of two phases that correspond to the two project tasks. Phase 1 will be the development of the TPA Beta Version 3.2 code. Phase 2 is the testing of the TPA Beta Version 3.2 code which will result in software changes that are needed for the final TPA Version 3.2 code.

4.3 Coding

All coding will be done in FORTRAN 77 with appropriate VAX extensions to accommodate the use

49/63

of long variable names. Coding style will be in accordance with that which has been historically used at the CNWRA and NRC for development of TPA codes.

4.4 Acceptance Testing and Analysis

As specified by NRC, the TPA Beta Version 3.2 must be able to be compiled, linked, and executed on CRADAL. CNWRA staff will ensure that these requirements are met prior to delivery. The results of these tests will be appropriately recorded in scientific notebooks or software change requests (SCR). All testing conducted to meet the Task 2 deliverable will be properly documented in scientific notebooks or SCRs. Changes made to the TPA Beta Version 3.2 during Task 2 will be properly documented prior to delivery of the TPA Version 3.2 code on July 17, 1998.

5.0 CONFIGURATION MANAGEMENT PLAN

The principal investigator (PI) for this project will implement suitable restrictions on access to the official version of the working code to ensure that coding conflicts do not arise during development. The methods used to restrict access will be defined by the PI. A copy of the final TPA Version 3.2 code will be provided to QA for configuration control. The TPA Beta Version 3.2 code will not be placed under formal configuration control because it will still be under joint NRC/CNWRA development at the time it is transmitted to the NRC staff on June 5, 1998. Note that this early transmittal of a Beta version is to facilitate further development and testing prior to completion and delivery of the code July 17, 1998.

6.0 REFERENCES

none

 5/27/98

7.0 APPENDICES

Appendix A

SOFTWARE REQUIREMENTS DESCRIPTION FOR TOTAL-SYSTEM PERFORMANCE ASSESSMENT CODE VERSION 3.2 May 7, 1998

This software requirements description (SRD) documents the modifications to be made in updating the Total-system Performance Assessment (TPA) code from Version 3.1.4 to 3.2. The original SRD for the TPA Code Version 3.0 can be found in appendix A of the TPA 3.0 User's Guide (Center for Nuclear Waste Regulatory Analyses, 1997). A series of Software Problem Correction Reports maintained in the Quality Assurance folder at the Center for Nuclear Waste Regulatory Analyses (CNWRA) documents changes subsequent to the SRD for TPA Code Version 3.0 leading to Version 3.1.4. The changes outlined in this SRD for Version 3.2 are proposed by the NRC and CNWRA staff to improve model conceptualization and flexibility in code application.

This SRD is broken down into two parts. Part I includes changes to the code that will increase flexibility of the code and Part II of the SRD outlines conceptual changes to various models. Part II of the SRD will be submitted after the changes are finalized at a series of key technical issues (KTI) meetings.

PART I

1.0 SOFTWARE FUNCTION

Two system-level changes are proposed to add flexibility to the latest Version of the TPA code: (i) allow code output at two compliance periods (e.g., 10,000 and 50,000 yr) time limits so that two sets of TPA runs will not be necessary and (ii) allow the user to specify a limited suite of important nuclides in order to expedite TPA runs.

The first change will allow the user to select two different output times and two time stepping procedures. This will allow the user to obtain outputs for a near-term compliance period (10,000 yr) while continuing simulations to a longer time period. Currently, the user may specify only one simulation period and the results are presented at the end of that time period. Consequently, if the user is interested in getting outputs for a time period different from the compliance period, two different TPA code executions must be conducted.

Two different time-stepping schemes are proposed so that calculations up to the compliance period can be done more precisely irrespective of the specified maximum simulation time. The time-stepping, for example, could be 125 uniform time steps of 80 yr from 0 to 10,000 yr and 75 increasing time steps from 10,000 to 50,000 yr.

The second change will permit the user specify only a selected groundwater nuclides. In the TPA Code Version 3.1.4, 20 nuclides for groundwater release must be tracked in order for the code to operate. The current code also requires the nuclides to be specified in a pre-specified order.

The user may also wish to analyze nuclides and decay chains other than those currently used in the TPA code above. The changes will allow the user to specify any decay chains and nuclides included in the INVENT utility module. However, the proposed modification will not include the capability of simulating two or more nuclides decaying to the same daughter nuclide. It is anticipated that modifications will be made to *tpa.inp*, the EBSREL, EXEC, READER, SZFT, and UZFT modules, the stand-alone code *releaset.f*, and the include files *maxnnucl.i* and *reader.i*.

All changes will be tested by running the TPA code. Results from Version 3.1.4 and the modified code will analyzed and compared.

2.0 TECHNICAL BASIS: PHYSICAL AND MATHEMATICAL MODEL

(not modified—see Center for Nuclear Waste Regulatory Analyses, 1997)

3.0 COMPUTATIONAL APPROACH

Please refer to the discussion in section 1.0

4.0 REFERENCES

Center for Nuclear Waste Regulatory Analyses, "Total-System Performance Assessment (TPA) Version 3.0 Code: Module Descriptions and User's Guide." March 1997.

APPENDICES

(none)

PART II

1.0 SOFTWARE FUNCTION

Far-field Radionuclide Transport

A joint NRC/CNWRA meeting was held on April 20, 1998 to determine improvements to radionuclide transport (RT) calculations in the TPA code. It was determined that the TPA code will not be modified for RT. However, the RT KTI team will provide input for updating matrix K_d s for each hydro-stratigraphic units in *tpa.inp* file. Due to the time constraints in the current TPA 3.2 development schedule, it was agreed that the KTI will provide new constraints for a limited number of sorption coefficient distributions for which sorption models have been calibrated. These will include distribution type, mean value, and standard deviations that can be added to the sorption coefficient distributions in the input file. It is also anticipated that correlations among sorption coefficients will be provided for those radionuclide K_d distributions that have been proposed to be modified.

Waste Package Failure Due to Seismic Events

Three general categories of modifications will be made to the TPA code for calculating waste package (WP) failure due to seismic events:

- A relationship (table or curve) between acceleration and fractional area of rock fall (known as the seismic heterogeneity factor in TPA 3.1.4) will be developed and implemented. The number of ground acceleration magnitude classes will be increased from 4 to 10. This change will replace the single "fractional area of rock fall" parameter in *tpa.inp* with 10 acceleration-dependent "fractional area of rock fall parameters". While all 10 parameters will be sampled for a particular realization, only those values that match the acceleration values for that realization will be utilized.
- The height of the yield zone (i.e., the rock column thickness) will be made a sampled parameter. In TPA 3.1.4 the height of the yield zone was specified as a constant in the *seismo.dat* data file and this constant value was picked corresponding to the acceleration-value and rock type. The proposed modification will replace each data point (i.e., the constant value) in *seismo.dat* with a probability distribution function from which the yield zone parameter will be sampled to allow the user to account for the uncertainty in the height of yield zone. This modification, however, will result in the elimination of the data file *seismo.dat* with each data point in this file being represented by an input parameter in *tpa.inp* that is either constant or sampled. This implementation will also remove limitations on the number of sampling points for seismic acceleration so that the effect of a wider range of ground accelerations can be assessed in the same TPA run. This option will make SEISMO more flexible. The change will introduce 50 new parameters in the TPA code assuming that 10 acceleration values instead of 4 will be used in TPA 3.2.

The above two modifications to SEISMO and the executive module were agreed upon at a joint NRC/CNWRA multi-KTI [Repository Design and Thermal-Mechanical Effects (RDTME), Structural Deformation and Seismicity (SDS), Container Life and Source Term, and Total System Performance Assessment and Integration] meeting held on April 30, 1998. A proposal was made to refine the failure criterion by taking into consideration energy consumption due to plastic deformation of WP material. This option could potentially remove some conservatism from the current SEISMO module. Ultimately it was decided that the failure criterion used in EBSFAIL (residual stress criterion) be used in SEISMO. However, this modification to SEISMO will be done only if it can be accommodated in the time frame for TPA 3.2.

Radionuclide Release from the Engineered Barrier System

A joint NRC/CNWRA meeting was held on April 16, 1998 to determine improvements to *releaset.f* stand-alone code and the TPA code to incorporate changes to the models and parameters affecting radionuclide releases from the engineered barrier system. The following changes were jointly agreed.

- Spent fuel (SF) leaching models in the EBSREL release module will be improved by adding the effect of secondary minerals. These changes are contingent upon the timely (yet to be determined) modifications by R. Codell, T. Ahn, and B. Murphy to the existing models. The formation of secondary mineral will alter the dissolution rate of the SF and will influence the release of radionuclides that are key contributors to dose.
- A provision will be made in EBSREL to account for changes in physical properties (porosity, permeability, and K_d) of the invert by the cementitious materials released from the liners. This will require modification of the *releaset.f* stand-alone code to account for advective flow through the backfill/invert material.
- Aqueous release of the gap fraction will be implemented in EBSREL. T. Ahn will collect data on gap/grain boundary inventory in SF. The gap fraction of the radionuclide inventory is important because it could be released instantaneously, thus influencing the peak dose.
- New parameters will be introduced to specify bathtub heights varying from subarea to subarea for corrosion, initially defective, and seismic failures. For faulting and volcanism, the bathtub height will be specified for the whole repository. This change will introduce additional variability in the radionuclide release calculations through randomizing the placement of the corrosion pit on the representative WP in each subarea.
- Capabilities will be added to evaluate release from WPs which have failed due to fracturing, on a time dependent basis (e.g., failure at 2000 yr, 5000 yr, 10,000 yr, and beyond 10,000 yr). This will replace the conservativeness in the current approach in which all WPs from scenario failures prior to corrosion failure are summed and assigned a failure time corresponding to the earliest event.

Effects such as the shrinking particle model and time dependent cladding failure will be done as separate evaluations and will not involve changing or using the TPA code. Off-line calculations will be conducted to account for the particle size distribution in SF dissolution. SF dry oxidation will not be tied to SF aqueous dissolution because the temperature will never exceed 250 °C above which powdered U_3O_8 will form. No code modifications will be made to account for grain boundary effects of SF in TPA 3.2

Near-Field Thermo-Hydrology and Chemistry

A joint NRC/CNWRA meeting was held on April 17, 1998. It was decided that the modifications to the REFLUX2 model that were proposed by the Thermal Effects on Flow (TEF) KTI to combine condensate flux in the fracture with the water driven out of the rock matrix, will replace the currently existing REFLUX 2 model in the TPA code. A suggestion was made to eliminate the REFLUX1 model; however, the group decided that REFLUX1 will continue to be a part of the TPA code as an alternative model.

Dose Conversion Factors

Dose conversion factors (DCFs) in DCAGW module will be updated to ensure internal (ingestion, inhalation) dose factors used in DCF calculations consistent with (or no less conservative than) dose factors applicable to soluble species used in groundwater transport calculations. Similarly, consistency/applicability of solubility assumptions used in DCAGS DCF calculations will be checked and DCFs will be updated, if needed. Though this activity will not result in any code changes, it may require substantial effort in running the GENII code to generate DCFs.

2.0 TECHNICAL BASIS: PHYSICAL AND MATHEMATICAL MODEL

Waste Package Failure Due to Seismic Events

To improve realism, CNWRA SDS and RDTME staff agreed that the TPA code should include a functional relationship between the level of earthquake-induced ground shaking (g level) and the fractional area of the repository that will undergo rock fall. This opinion is supported by limited data of the effects of earthquakes on underground facilities (e.g., Sharma and Judd, 1991). The approach to develop this functional relationship is through informal elicitation of CNWRA staff. Eleven CNWRA staff participated in the development of the g-fractional area relationship. Results are currently being integrated by B. Sagar and will be forthcoming. The TPA code will be modified as stated in the previous section to accommodate this addition.

Radionuclide Release from the Engineered Barrier System

Technical bases for the changes are evident in the previous section.

Near-Field Thermo-Hydrology and Chemistry

To improve realism, CNWRA TEF staff agreed that the refluxing modules (REFLUX1 And REFLUX2) in the TPA code should contain a greater degree of realism. REFLUX2 was modified to include a procedure to estimate the depth that water may penetrate below the boiling isotherm. The modified REFLUX2 will include both infiltration and ambient rock water as the source for the refluxing water.

Dose Conversion Factors

Technical bases for the changes are evident in the previous section.

3.0 COMPUTATIONAL APPROACH

(not modified—see Center for Nuclear Waste Regulatory Analyses, 1997)

4.0 REFERENCES

Center for Nuclear Waste Regulatory Analyses, "Total-System Performance Assessment (TPA) Version 3.0 Code: Module Descriptions and User's Guide." March 1997.

Sharma, S., and W.R. Judd. 1991. Underground opening damage from earthquakes. *Engineering Geology* 30: 263-276.

APPENDICES

(none)

APPROVED:

Signature of Element Manager

Date

Appendix B

Gordon,

To facilitate the TPA Version 3.2 testing and production process, the following criteria are NRC's expectation on the two deliverables:

1) For TPA 3.2 Beta Version (due on June 5, 1998) - This version of the code will be used to evaluate the correctness of the changes (as stated in the TPA 3.2 SRD) which implies we need to be able to run the code with an input file and check the correctness of the *.res files. The June deliverable is acceptable if the code can be compiled on NRC's computing system, executed with the supplied input file, and successfully produce the *.res files. Correctness of the *.res files is desired but not required for acceptability of the Beta version.

2) For TPA Version 3.2 (due on July 17, 1998) - This version of the code will be used to evaluate the VA and conduct sensitivity analyses. The July deliverable is acceptable if the code can be compiled on NRC's computing system, executed with the supplied input file, and the *.res files are correct. All corrections identified from our joint testing effort should also be implemented.

Hopefully the above criteria provide you a more clear idea on how we will evaluate the acceptability of the two deliverables. Please contact Tim McCartin or me if you have any further concern regarding the TPA code.

Christiana

REQUIREMENT DESCRIPTION

**SOFTWARE REQUIREMENTS DESCRIPTION
FOR TOTAL-SYSTEM PERFORMANCE ASSESSMENT CODE VERSION 3.2
May 7, 1998**

This software requirements description (SRD) documents the modifications to be made in updating the Total-system Performance Assessment (TPA) code from Version 3.1.4 to 3.2. The original SRD for the TPA Code Version 3.0 can be found in appendix A of the TPA 3.0 User's Guide (Center for Nuclear Waste Regulatory Analyses, 1997). A series of Software Problem Correction Reports maintained in the Quality Assurance folder at the Center for Nuclear Waste Regulatory Analyses (CNWRA) documents changes subsequent to the SRD for TPA Code Version 3.0 leading to Version 3.1.4. The changes outlined in this SRD for Version 3.2 are proposed by the NRC and CNWRA staff to improve model conceptualization and flexibility in code application.

This SRD is broken down into two parts. Part I includes changes to the code that will increase flexibility of the code and Part II of the SRD outlines conceptual changes to various models. Part II of the SRD will be submitted after the changes are finalized at a series of key technical issues (KTI) meetings.

PART I

1.0 SOFTWARE FUNCTION

Two system-level changes are proposed to add flexibility to the latest Version of the TPA code: (i) allow code output at two compliance periods (e.g., 10,000 and 50,000 yr) time limits so that two sets of TPA runs will not be necessary and (ii) allow the user to specify a limited suite of important nuclides in order to expedite TPA runs.

The first change will allow the user to select two different output times and two time stepping procedures. This will allow the user to obtain outputs for a near-term compliance period (10,000 yr) while continuing simulations to a longer time period. Currently, the user may specify only one simulation period and the results are presented at the end of that time period. Consequently, if the user is interested in getting outputs for a time period different from the compliance period, two different TPA code executions must be conducted.

Two different time-stepping schemes are proposed so that calculations up to the compliance period can be done more precisely irrespective of the specified maximum simulation time. The time-stepping, for example, could be 125 uniform time steps of 80 yr from 0 to 10,000 yr and 75 increasing time steps from 10,000 to 50,000 yr.

The second change will permit the user specify only a selected groundwater nuclides. In the TPA Code Version 3.1.4, 20 nuclides for groundwater release must be tracked in order for the code to operate. The current code also requires the nuclides to be specified in a pre-specified order.

The user may also wish to analyze nuclides and decay chains other than those currently used in the TPA code above. The changes will allow the user to specify any decay chains and nuclides included in the INVENT utility module. However, the proposed modification will not include the capability of simulating two or more nuclides decaying to the same daughter nuclide. It is anticipated that modifications will be made

to *tpa.inp*, the EBSREL, EXEC, READER, SZFT, and UZFT modules, the stand-alone code *releaset.f*, and the include files *maxnucl.i* and *reader.i*.

All changes will be tested by running the TPA code. Results from Version 3.1.4 and the modified code will be analyzed and compared.

2.0 TECHNICAL BASIS: PHYSICAL AND MATHEMATICAL MODEL

(not modified—see Center for Nuclear Waste Regulatory Analyses, 1997)

3.0 COMPUTATIONAL APPROACH

Please refer to the discussion in section 1.0

4.0 REFERENCES

Center for Nuclear Waste Regulatory Analyses, "Total-System Performance Assessment (TPA) Version 3.0 Code: Module Descriptions and User's Guide." March 1997.

APPENDICES

(none)

PART II

1.0 SOFTWARE FUNCTION

Far-field Radionuclide Transport

A joint NRC/CNWRA meeting was held on April 20, 1998 to determine improvements to radionuclide transport (RT) calculations in the TPA code. It was determined that the TPA code will not be modified for RT. However, the RT KTI team will provide input for updating matrix K_d s for each hydro-stratigraphic units in *tpa.inp* file. Due to the time constraints in the current TPA 3.2 development schedule, it was agreed that the KTI will provide new constraints for a limited number of sorption coefficient distributions for which sorption models have been calibrated. These will include distribution type, mean value, and standard deviations that can be added to the sorption coefficient distributions in the input file. It is also anticipated that correlations among sorption coefficients will be provided for those radionuclide K_d distributions that have been proposed to be modified.

Waste Package Failure Due to Seismic Events

Three general categories of modifications will be made to the TPA code for calculating waste package (WP) failure due to seismic events:

- A relationship (table or curve) between acceleration and fractional area of rock fall (known as the seismic heterogeneity factor in TPA 3.1.4) will be developed and implemented. The number of ground acceleration magnitude classes will be increased from 4 to 10. This

change will replace the single "fractional area of rock fall" parameter in *tpa.inp* with 10 acceleration-dependent "fractional area of rock fall parameters". While all 10 parameters will be sampled for a particular realization, only those values that match the acceleration values for that realization will be utilized.

- The height of the yield zone (i.e., the rock column thickness) will be made a sampled parameter. In TPA 3.1.4 the height of the yield zone was specified as a constant in the *seismo.dat* data file and this constant value was picked corresponding to the acceleration-value and rock type. The proposed modification will replace each data point (i.e., the constant value) in *seismo.dat* with a probability distribution function from which the yield zone parameter will be sampled to allow the user to account for the uncertainty in the height of yield zone. This modification, however, will result in the elimination of the data file *seismo.dat* with each data point in this file being represented by an input parameter in *tpa.inp* that is either constant or sampled. This implementation will also remove limitations on the number of sampling points for seismic acceleration so that the effect of a wider range of ground accelerations can be assessed in the same TPA run. This option will make SEISMO more flexible. The change will introduce 50 new parameters in the TPA code assuming that 10 acceleration values instead of 4 will be used in TPA 3.2.

The above two modifications to SEISMO and the executive module were agreed upon at a joint NRC/CNWRA multi-KTI [Repository Design and Thermal-Mechanical Effects (RDTME), Structural Deformation and Seismicity (SDS), Container Life and Source Term, and Total System Performance Assessment and Integration] meeting held on April 30, 1998. A proposal was made to refine the failure criterion by taking into consideration energy consumption due to plastic deformation of WP material. This option could potentially remove some conservatism from the current SEISMO module. Ultimately it was decided that the failure criterion used in EBSFAIL (residual stress criterion) be used in SEISMO. However, this modification to SEISMO will be done only if it can be accommodated in the time frame for TPA 3.2.

Radionuclide Release from the Engineered Barrier System

A joint NRC/CNWRA meeting was held on April 16, 1998 to determine improvements to *releaset.f* stand-alone code and the TPA code to incorporate changes to the models and parameters affecting radionuclide releases from the engineered barrier system. The following changes were jointly agreed.

- Spent fuel (SF) leaching models in the EBSREL release module will be improved by adding the effect of secondary minerals. These changes are contingent upon the timely (yet to be determined) modifications by R. Codell, T. Ahn, and B. Murphy to the existing models. The formation of secondary mineral will alter the dissolution rate of the SF and will influence the release of radionuclides that are key contributors to dose.
- A provision will be made in EBSREL to account for changes in physical properties (porosity, permeability, and K_d) of the invert by the cementitious materials released from the liners. This will require modification of the *releaset.f* stand-alone code to account for advective flow through the backfill/invert material.

- Aqueous release of the gap fraction will be implemented in EBSREL. T. Ahn will collect data on gap/grain boundary inventory in SF. The gap fraction of the radionuclide inventory is important because it could be released instantaneously, thus influencing the peak dose.
- New parameters will be introduced to specify bathtub heights varying from subarea to subarea for corrosion, initially defective, and seismic failures. For faulting and volcanism, the bathtub height will be specified for the whole repository. This change will introduce additional variability in the radionuclide release calculations through randomizing the placement of the corrosion pit on the representative WP in each subarea.
- Capabilities will be added to evaluate release from WPs which have failed due to fracturing, on a time dependent basis (e.g., failure at 2000 yr, 5000 yr, 10,000 yr, and beyond 10,000 yr). This will replace the conservativeness in the current approach in which all WPs from scenario failures prior to corrosion failure are summed and assigned a failure time corresponding to the earliest event.

Effects such as the shrinking particle model and time dependent cladding failure will be done as separate evaluations and will not involve changing or using the TPA code. Off-line calculations will be conducted to account for the particle size distribution in SF dissolution. SF dry oxidation will not be tied to SF aqueous dissolution because the temperature will never exceed 250 °C above which powdered U_3O_8 will form. No code modifications will be made to account for grain boundary effects of SF in TPA 3.2

Near-Field Thermo-Hydrology and Chemistry

A joint NRC/CNWRA meeting was held on April 17, 1998. It was decided that the modifications to the REFLUX2 model that were proposed by the Thermal Effects on Flow (TEF) KTI to combine condensate flux in the fracture with the water driven out of the rock matrix, will replace the currently existing REFLUX 2 model in the TPA code. A suggestion was made to eliminate the REFLUX1 model; however, the group decided that REFLUX1 will continue to be a part of the TPA code as an alternative model.

Dose Conversion Factors

Dose conversion factors (DCFs) in DCAGW module will be updated to ensure internal (ingestion, inhalation) dose factors used in DCF calculations consistent with (or no less conservative than) dose factors applicable to soluble species used in groundwater transport calculations. Similarly, consistency/applicability of solubility assumptions used in DCAGS DCF calculations will be checked and DCFs will be updated, if needed. Though this activity will not result in any code changes, it may require substantial effort in running the GENII code to generate DCFs.

2.0 TECHNICAL BASIS: PHYSICAL AND MATHEMATICAL MODEL

Waste Package Failure Due to Seismic Events

To improve realism, CNWRA SDS and RDTME staff agreed that the TPA code should include a functional relationship between the level of earthquake-induced ground shaking (g level) and the fractional area of the repository that will undergo rock fall. This opinion is supported by limited data of the effects of earthquakes on underground facilities (e.g., Sharma and Judd, 1991). The approach to develop this functional

relationship is through informal elicitation of CNWRA staff. Eleven CNWRA staff participated in the development of the g-fractional area relationship. Results are currently being integrated by B. Sagar and will be forthcoming. The TPA code will be modified as stated in the previous section to accommodate this addition.

Radionuclide Release from the Engineered Barrier System

Technical bases for the changes are evident in the previous section.

Near-Field Thermo-Hydrology and Chemistry

To improve realism, CNWRA TEF staff agreed that the refluxing modules (REFLUX1 And REFLUX2) in the TPA code should contain a greater degree of realism. REFLUX2 was modified to include a procedure to estimate the depth that water may penetrate below the boiling isotherm. The modified REFLUX2 will include both infiltration and ambient rock water as the source for the refluxing water.

Dose Conversion Factors

Technical bases for the changes are evident in the previous section.

3.0 COMPUTATIONAL APPROACH

(not modified—see Center for Nuclear Waste Regulatory Analyses, 1997)

4.0 REFERENCES

Center for Nuclear Waste Regulatory Analyses, "Total-System Performance Assessment (TPA) Version 3.0 Code: Module Descriptions and User's Guide." March 1997.

Sharma, S., and W.R. Judd. 1991. Underground opening damage from earthquakes. *Engineering Geology* 30: 263-276.


APPENDICES

(none)

APPROVED:


Signature of Element Manager

5/7/98
Date

TPA V 3.2 PC 

1/88

SOFTWARE RELEASE NOTICE

| | | |
|--|----------------------|--------------------------|
| 1. SRN Number: PA-SRN-212 | | |
| 2. Project Title: TSPA & Technical Integration Code | | Project No. 20-01402-762 |
| 3. SRN Title: TPA Version 3.2 PC B Rg 10-4-99 | | |
| 4. Originator/Requestor: Ron Janetzke | | Date: Oct. 1, 1999 |
| 5. Summary of Actions <ul style="list-style-type: none"> <input type="checkbox"/> Release of new software <input type="checkbox"/> Release of modified software: <ul style="list-style-type: none"> <input type="checkbox"/> Enhancements made <input type="checkbox"/> Corrections made <input type="checkbox"/> Change of access software <input checked="" type="checkbox"/> Software Retirement | | |
| 6. Persons Authorized Access | | |
| Name | Read Only/Read-Write | Addition/Change/Delete |
| Ron Janetzke | RW | |
| Sitakanta Mohanty | RW | |
| Tim McCartin (NRC) | RW | |
| M. Rose Byrne (NRC) | RW | |
| 7. Element Manager Approval: <i>Anders Wittmeyer</i> | | Date: 10/4/99 |
| 8. Remarks: Version superceded. | | |

2/88

SOFTWARE SUMMARY FORM

| | | | |
|---|---|--|---|
| 01. Summary Date: 12/1/98 | 02. Summary prepared by (Name and phone): <i>SM</i> STEP MICHAEL MULLER 12/1/98 | | 03. Summary Action: NEW |
| 04. Software Date: 12/4/98 | 05. Short Title: TPA Version 3.2 PCβ | | |
| 06. Software Title: TPA-System Performance Assessment Computer Code, Version 3.2 PC BETA | | | 07. Internal Software ID: None |
| 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 type="checkbox"/> Scientific/Engineering <input type="checkbox"/> Auxiliary Analyses <input checked="" type="checkbox"/> Total System PA <input type="checkbox"/> Subsystem PA <input type="checkbox"/> Other b. Specific: | |
| 11. Submitting Organization and Address: CNWRA/SwRI 6220 Culebra Road San Antonio, TX 78228 | | 12. Technical Contact(s) and Phone: Sitakanta Mohanty (210) 522-5185 Michael Muller (210) 522-2296 | |
| 13. Software Application: The TPA Code consists of the following modules: UZFLOW, NFENV, EBSREL, UZFT, SZFT, DCAGW, FAULTO, SEISMO, VOLCANO, ASHPLUMO, ASHRMVO, DCAGS, LHS, EXEC. | | | |
| 14. Computer Platform: SUN Workstation PC | 15. Computer Operating System: UNIX for SUN NT for PCs | 16. Programming Language(s): FORTRAN | 17. Number of Source Program Statements: Approx. 39,700 lines w/o Stand Alone Codes |
| 18. Computer Memory Requirements: 45 Mb | 19. Tape Drives: N/A | 20. Disk Units: N/A | 21. Graphics: N/A |
| 22. Other Operational Requirements: None | | | |
| 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"/> Preliminary <input type="checkbox"/> In-House ONLY | |
| 25. <i>Sitakanta Mohanty</i> Software Developer: | | Date: 12/1/98 | |

SOFTWARE RELEASE NOTICE

| | | |
|---|----------------------|--------------------------------|
| 1. SRN Number: PA-SRN-184 | | |
| 2. Project Title: TPA Version 3.2 PCβ | | Project No. 20-1402-762 |
| 3. SRN Title: TPA Version 3.2 PCβ | | |
| 4. Originator/Requestor: Bruce Mabrito | | Date: 12/1/98 |
| 5. Summary of Actions <input checked="" 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 type="checkbox"/> Software Retirement | | |
| 6. Persons Authorized Access | | |
| Name | Read Only/Read-Write | Addition/Change/Delete |
| Sitakanta Mohanty | RW | Addition |
| Tim McCartin (NRC) | RW | Addition |
| M. Rose Byrne (NRC) | RW | Addition |
| 7. Element Manager Approval: <i>Gordon Whitmeyer</i> Date: <i>12/1/98</i> | | |
| 8. Remarks: | | |

**PROCEDURE FOR INSTALLING AND MODIFYING
TOTAL-SYSTEM PERFORMANCE ASSESSMENT (TPA)
VERSION 3.2PCβ CODE**

Prepared for

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

Prepared by

**Michael Muller
Mike Epley
Sitakanta Mohanty**

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

December 1998

ACKNOWLEDGMENTS

This work was performed on behalf of the Nuclear Regulatory Commission (NRC) Office of Nuclear Material Safety and Safeguards, Division of Waste Management under contract No. NRC-02-97-009. This document is an independent product of the Center for Nuclear Waste Regulatory Analyses (CNWRA) and does not necessarily reflect the views or regulatory position of the NRC.

The TPA 3.2PC β code has been developed following the procedures described in the CNWRA Technical Operating Procedure, TOP-018, which implements the quality assurance (QA) guidance contained in the CNWRA QA Manual.

The authors thank Ron Janetzke and Wesley Patrick for reviews of this report. The authors are thankful to Cathy Garcia for typing and formatting help in preparing the document.

CONTENTS

| Section | Page |
|--|------|
| ACKNOWLEDGMENTS | iii |
| TABLES | v |
| 1 INTRODUCTION | 1 |
| 2 INSTALLATION AND EXECUTION | 1 |
| 2.1 DISTRIBUTION OF TPA3.2PC β SOFTWARE | 1 |
| 2.2 INSTALLING TPA3.2PC β CODE FROM FLOPPY DISKETTES | 1 |
| 2.2.1 Procedures for Installing the Standard/Uncompiled Code | 2 |
| 2.2.2 Procedure for Installing the Precompiled Code | 3 |
| 2.3 PROGRAM COMPILATION | 4 |
| 2.4 PROGRAM EXECUTION | 4 |
| 2.5 SYSTEM REQUIREMENTS | 6 |
| 3 CODE VERIFICATION | 7 |
| 4 USER SUPPORT | 7 |
| 5 REFERENCES | 8 |
| APPENDIX A DETAILS FOR CODE DEVELOPERS | |

TABLES

| Table | | Page |
|-------|---|------|
| 2-1 | Actual memory and disk space requirements for TPA3.2PC β basecase | 8 |
| 2-2 | Run time comparison using TPA3.2PC β basecase | 8 |

1 INTRODUCTION

The primary use of the Total-system Performance Assessment (TPA) code is to estimate the expected radionuclide dose from the proposed Yucca Mountain high-level radioactive waste repository for specified time periods and at designated receptor group locations. Version 3.2 of the TPA code was developed on a Sun Microsystems, Inc. SPARC 20 computer. The code runs on any platform using the Sun Microsystems UNIX operating system (Solaris) and the Sun FORTRAN 77 compiler. This document describes the development and implementation of a version of the code that has been ported to a personal computer (PC) running Microsoft Windows NT 4.0 operating system (NT) with the Lahey FORTRAN 90 compiler. The ported code is referred to as TPA3.2PC β . The same code can now be compiled and executed on both UNIX-Solaris and PC-NT platforms.

The installation and execution of TPA3.2PC β are described in section 2. Verification of the code is described in section 3. Porting issues of interest to code developers making modifications to the code are described in appendix A.

2 INSTALLATION AND EXECUTION

This section describes the procedure for installation and execution of TPA3.2PC β on a PC running NT. It is assumed the user is familiar with running the TPA Version 3.2 code as described in the TPA Version 3.2 user's guide (Mohanty and McCartin, 1998).

2.1 DISTRIBUTION OF TPA3.2PC β SOFTWARE

The TPA3.2PC β code consists of many files arranged into one main directory and four subdirectories. The majority of the files are FORTRAN source and *include* files as well as the requisite data files for TPA3.2PC β . Also included are utility files to assist in compiling and executing the program.

These files are delivered on CD ROM, 8mm tape, or two diskettes. The standard distribution contains only the FORTRAN source files. There is also a precompiled version (i.e., does not require Lahey FORTRAN) available on three diskettes. The installation procedures are described in the next section.

2.2 INSTALLING TPA3.2PC β CODE FROM FLOPPY DISKETTES

This section describes an installation from diskettes. The procedure is identical for the TPA Version 3.2 code distributed on CD ROM but slightly different for the 8mm distribution. TPA3.2PC β consists of many files arranged into one main directory and four subdirectories (i.e., *ccdf*, *codes*, *data*, and *portpc*) as shown:

```
tpa\ccdf
tpa\codes
tpa\data
tpa\portpc
```

The main directory, *tpa*, contains the FORTRAN source (*.f) and *include* (*.i) files for the main TPA code. The subdirectory, *ccdf*, contains the auxiliary codes. The subdirectory, *codes*, contains the source code

for the six standalone support codes that the main TPA code runs (which can also be run by themselves). The subdirectory, *data*, contains precalculated data files in ASCII text format. The subdirectory, *portpc*, contains helper files and utilities that may be useful for the TPA Version 3.2 code on a PC.

The procedure given in section 2.2.1 describes installation of the standard/uncompiled version that requires Lahey FORTRAN 90 Version 4.5. If the Lahey compiler is not available, the user should refer to section 2.2.2.

All files comprising TPA3.2PC β are compressed into two files, *tpa.exe* and *data.exe*. They can be uncompressed in any order as long as performed in the proper directories. The files are compressed in a standard self-extracting ZIP format that both unpacks and installs the files on execution of the ZIP utility.

The instructions in the next sentence assume that Lahey FORTRAN is installed and that *lf9045\bin* is in the user's path. If it is required to add the *lf9045\bin* directory, type: `set path=%path%;c:\lf9045\bin`. It is also assumed that an NT command prompt/DOS window is open for use.

2.2.1 Procedures for Installing the Standard/Uncompiled Code

To install the TPA3.2PC β code, follow this procedure:

- Create a top-level directory and enter that directory. The sample names, *cnwra* and *tpa* can be changed.

```
mkdir c:\cnwra
mkdir c:\cnwra\tpa
cd c:\cnwra\tpa
```

- Copy the contents of each diskette to the current directory.

```
Insert diskette 1, copy a:\*.* .
Insert diskette 2, copy a:\*.* .
```

- Enter the *tpa.exe* command. The *tpa.exe* command will unpack the source code files for the main *tpa* directory and its subdirectories from the section of the *tpa.exe* file containing the compressed information. It then creates the *ccdf*, *codes*, and *portpc* subdirectories and places the code files into these subdirectories.

```
tpa.exe
```

- Create and unZIP the *data* subdirectory and files. This creates the *data* subdirectory and places the data files into that subdirectory.

```
mkdir data
cd data
..\data.exe
```

- Remove the **.exe* (ZIP) files (not mandatory).

```
del ..\tpa.exe
del ..\data.exe
```

Now the directory structure shown in section 2.2 should be complete and there should be **f* files in the *tpa*, *ccdf*, and *codes* directories and **.dat* files in the *data* subdirectory. The DOS `dir` command can be used to confirm this. If the directory structure is incorrect or the files are absent, the previous installation procedure should be repeated. The program can then be compiled and executed.

2.2.2 Procedure for Installing the Precompiled Code

To install the TPA3.2PC β precompiled code files, follow this procedure:

- Create a top-level directory and enter that directory. The sample names, *cnwra* and *tpa* can be changed.

```
mkdir c:\cnwra
mkdir c:\cnwra\tpa
cd c:\cnwra\tpa
```

- Copy the contents of each diskette to the current directory.

```
Insert diskette 1, copy a:\*.* .
Insert diskette 2, copy a:\*.* .
Insert diskette 3, copy a:\*.* .
```

- UnZIP the code for the main *tpa* directory and its subdirectories from the file *tpa.exe*. This unpacks the main *tpa* files, creates the *ccdf* and *portpc* subdirectories, and places code files into these subdirectories.

```
tpa.exe
```

- Create and unZIP the *data* subdirectory and files. This creates the *data* subdirectory and places the data files into that subdirectory.

```
mkdir data
cd data
..\data.exe
```

- Create and unZIP the *codes* subdirectory and files. This creates the *codes* subdirectory and puts the data files into that subdirectory.

```
cd ..
mkdir codes
cd codes
..\codes.exe
```

- Remove the *.exe* (ZIP) files (not mandatory).

```
del ..\tpa.exe
del ..\data.exe
del ..\codes.exe
```

Now the directory structure shown in section 2.2 should be complete and there should be **.f* files in the *tpa*, *ccdf*, and *codes* directories and **.dat* files in the *data* subdirectory. The DOS *dir* command can be used to confirm this. If the directory structure is incorrect or the files are absent, the previous installation procedure should be repeated. With installation complete, the program, which is already compiled, can be run.

2.3 PROGRAM COMPILATION

Once the source code for TPA3.2PC β is installed, it can be compiled by following these instructions:

```
cd c:\cnwra\tpa
```

This will change the working directory to the main TPA directory. Along with the source code, it contains: *Makefile*, *make.bat*, and many *.fig* files. The file, *Makefile*, is for UNIX use only. The file, *make.bat*, is for PC use and calls Lahey's AUTOMAKE facility, which uses the commands stored in the *automake.fig* file. To compile the code, type

```
make
```

The Lahey compiler will execute and begin to generate many screen messages, most of which can be ignored. Compilation typically requires 15 minutes on the recommended system described in section 2.5.

For subsequent compilation (after editing source files), simply retype: *make*. This command will compile and link the *exec.f* (*tpa* main) file with the appropriate *.obj* files. If it becomes necessary to start over, delete all *dependency* (**.dep*) and *object* (**.obj*) files except *lfsystem.obj*, then type: *make*. Like UNIX *make*, AUTOMAKE ensures that each **.obj* file is up to date. AUTOMAKE checks the date and time of each source file. If any source (**.f*) or include (**.i*) files have been changed since the **.obj* files were made, they are recompiled to generate new **.obj* files, then linked with the other **.obj* files. Only those FORTRAN source files that have been changed are re-compiled. The final executable program is called *tpa.exe*.

Compilation will stop with a *Quitting* error if a fatal error occurs. The most common fatal errors are due to insufficient disk space. For other errors, consult a FORTRAN manual. If the errors cannot be resolved, the support staff listed in section 4 can be consulted. After the errors are fixed, delete the appropriate **.obj* files and type: *make* again. The file, *lfsystem.obj*, must be in the main *tpa* directory for compilation. If it is accidentally deleted, it can be copied from the *portpc* directory.

2.4 PROGRAM EXECUTION

The only directory names that cannot be changed are *codes* and *data*. The TPA code can be run from any directory provided the following procedure is used.

- Set the TPA environment variables to point to the parent directory of the *codes* and *data* subdirectories. The TPA code uses these variables to access the auxiliary input data and standalone programs in these subdirectories. It is important to remember that the path name *cnwra\tpa* is an example; the actual path used for installation on a particular PC should be replaced in the commands listed. There are two ways to set the system variables. The first way applies only to the DOS window into which it is typed. The second way applies to every DOS window opened after it is implemented.

— for the current DOS window only

```
set TPA_DATA=c:\cnwra\tpa
set TPA_TEST=c:\cnwra\tpa
```

— for each new DOS window opened in the future

Using the mouse, open the *Control Panel*; open the *System* program; click on the *Environment* tab. At the bottom of the dialog box, there are two fields for typing in the environment variable and its value.

Click on the *Variable* field and type

TPA_DATA

Then click on the *Value* field (or use Tab key) and type

c:\cnwra\tpa

Then click on the *Set* button to activate the change.

Then enter the second environment variable and value.

Click on the *Variable* and type

TPA_TEST

Then click on the *Value* field (or use Tab key) and type

c:\cnwra\tpa

Then click the *Set* button to activate the change.

Close the window by clicking on the *OK* button.

One can check that the environment variables have been set correctly by opening a new DOS window and typing

```
dir %TPA_DATA%\data      (there should be many *.dat files)
dir %TPA_TEST%\codes     (there should be many *.f and *.fig files)
```


Typing the DOS `set` command by itself shows the current values of all environment variables.

- Next, make a directory for code execution. This directory needs a minimum of 130 megabytes (MB) disk space for a basecase run.

```
mkdir c:\cnwra\run1
```

Enter that directory and copy *tpa.inp* into it:

```
cd c:\cnwra\run1
copy c:\cnwra\tpa\tpa.inp .
```

- If Lahey FORTRAN is installed, proceed to the next bullet. Otherwise, move the file *lf90.eer* to the current directory by typing

```
copy c:\cnwra\tpa\portpc\lf90.eer .
```

- The code is now ready to run. The executable *tpa.exe* may be copied into the current directory but that is not mandatory. Only *tpa.inp* is required in the current directory. TPA3.2PC β prints many messages to the screen as it runs. If these are to be saved to a file, the DOS redirection command, `>`, should be used as it is in UNIX.

To run with output directed to the screen, type

```
c:\cnwra\tpa\tpa.exe
```

To run with output directed to a file, type

```
c:\cnwra\tpa\tpa.exe > tpa.out
```

The DOS window recognizes *tpa.exe* or *tpa* (with an assumed *.exe* extension) as a program to run. In the example, the file, *tpa.out*, will hold all output that TPA3.2PC β normally writes to the screen.

2.5 SYSTEM REQUIREMENTS

The following system requirements are recommended based on the operating systems and FORTRAN compilers that were used during the testing phase of the code:

- Windows NT 4.0 servicepac3
- Lahey FORTRAN 90 Version 4.5e + *lfsystem.obj patch* file included in the distribution

The following basic hardware requirements are recommended based on the experience gained during the testing phase of the code:

- 200 MHz Pentium class processor
- 128 MB physical memory

- 1 GB hard disk
- 400 MB of virtual memory (NT)

The recommended basic hardware is the minimum system configuration on which the TPA 3.2PC β code can run in a reasonable time. The code will run on systems with less than the basic hardware but compilation/execution times will degrade sharply. Total hard disk space requirements, beyond the minimum actually used (262 MB) for testing the code, are discussed in detail in the TPA Version 3.2 user's guide (Mohanty and McCartin, 1998). Also note that even with the basic system configuration, a machine running NT will respond slowly if any other program is run concurrently. UNIX does not exhibit this problem. To decrease execution times under NT, the system configuration should be modified by adding in order of effectiveness: (i) a faster processor, (ii) a faster disk/disk controller/driver, and (iii) more memory (160 MB is probably the maximum needed for dedicated execution of TPA3.2PC β). The memory used by TPA3.2PC β under NT 4.0 servicepac3 with 400 MB virtual memory is presented in table 2-1. A comparison of run times is presented in table 2-2. The longer run time per realization for the 10-realization run compared to the 300-realization run is primarily due to the initial time spent loading the code into memory.

3 CODE VERIFICATION

The code was verified by comparing TPA3.2PC β output with TPA3.2 output on a Sun. When comparing the output of several runs of both codes on a Sun Solaris workstation, all files were identical. This was expected as the modifications were designed to enable the code to run under NT on a PC without impacting the execution of the software when compiled under UNIX.

The code was also verified by comparing TPA3.2PC β output on a UNIX platform and TPA3.2PC β output on an NT platform. The output from each standalone code under NT was compared to the output under UNIX—the files with formatted output were identical. While the exact formats differed, the numerical values in the files with free format were equal within the presented precision. When the full TPA code output was compared for NT and UNIX, many differences were found. These differences, described in appendix A, are compiler dependent and not a result of incorrect coding. Thus, with minor modifications to the code, the NT and UNIX versions of the TPA code can be made platform-independent.

4 USER SUPPORT

For technical assistance, users may contact

Sitakanta Mohanty
Center for Nuclear Waste Regulatory Analyses
Southwest Research Institute
P.O. Drawer 28510
San Antonio, TX 78228-0510
(210) 522-5185
smohanty@swri.edu

Michael Muller
Southwest Research Institute
P.O. Drawer 28510
San Antonio, TX 78228-0510
(210) 522-3296
mmuller@swri.edu

Table 2-1. Actual memory and disk space requirements for TPA3.2PC β basecase

| Type | Values |
|---|--|
| Physical memory (not virtual memory) | 35 MB for NT OS and 65 MB for TPA3.2PC β |
| Hard disk file space (not virtual memory) | 6 MB compiled main directory, 5 MB <i>data</i> , 120 MB compiled <i>codes</i> directory, and 131 MB base case run (262 MB total) |

Table 2-2. Run time comparison using TPA3.2PC β basecase

| Processor | RAM (MB) | Realizations | Run time (minutes) |
|--------------------------------------|----------|--------------|--------------------|
| P200 | 128 | 10 | 20 |
| PII300 | 265 | 10 | 13 |
| P200 | 128 | 300 | 425 |
| PII300 | 265 | 300 | 253 |
| P200 --> Intel Pentium, 200 MHz | | | |
| PII300 --> Intel Pentium II, 300 MHz | | | |

5 REFERENCES

Mohanty, S., and T.J. McCartin. 1998. *Total-system Performance Assessment (TPA) Version 3.2 Code: Module Descriptions and User's Guide*. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

APPENDIX A

A. DETAILS FOR CODE DEVELOPERS

This appendix discusses the details of changes needed to enable the execution of the TPA Version 3.2 code on the PC platform using Lahey FORTRAN 90. The discussions encompass program compilation, program execution/errors, NT/DOS filenames, and portability issues (UNIX versus NT).

Lahey FORTRAN bugs encountered during the code compilation, memory and disk space requirements, and the Lahey compiler options are presented in section A.1. Code execution related issues and associated errors are presented in section A.2. Restrictions pertaining to length of filenames under NT on a PC platform are discussed in section A.3. Changes to FORTRAN statements to make the code portable from Sun to PC platforms and vice versa are presented in section A.4.

A.1 PROGRAM COMPILATION

The Lahey FORTRAN 90 (*lf90*) Versions 4.0 and 4.5 compilers have two bugs in the *system* subroutine. One bug involves a failure to close Windows object handles. NT does not do this automatically, while Windows 95 does. It is unknown how Windows 98 resolves this. The other bug involves failure of the redirection command, *>*, when used by a child process after the parent process screen output has already been redirected. The child process redirection fails under NT and Windows 95. It is not clear at this point how Windows 98 resolves the redirection problem.

A special file *lfsystem.obj* has been included with the TPA2.3PC β distribution to repair these bugs. This file, *lfsystem.obj*, contains the *system* subroutine, which substitutes for the one in the Lahey FORTRAN function library. This substitute subroutine works with *lf90* Version 4.5e. It does not work with *lf90* version 4.0. The Lahey compiler, *lf90* version 5.0, is reputed to work correctly with or without the *lfsystem.obj* file. This file resides in, and is linked from, the main *tpa* directory. A copy exists in the *portpc* subdirectory as a backup copy in case the file in the main directory is accidentally deleted.

If the program is compiled without *lfsystem.obj*, two problems will be experienced under NT: (i) handles will be used up until a screen error occurs with a pop-up window (this is at about 50 realizations for a 128 MB memory machine) and (ii) screen output from the standalone programs will be lost. These limitations can be annoying but are not fatal.

The *tpa.exe* program will compile the same whether or not the *lfsystem.obj* file exists because the *system* routine from the Lahey FORTRAN library will be linked in and used or needed. Using the batch file command *make* to create the standalone programs will generate a *file not found* error if the *lfsystem.obj* file does not exist. This error notification is due to the explicit use of the file, *lfsystem.obj*, in the standalone *make* file, *makeall.bat*. To remedy this, the file, *lfsystem.obj*, can be copied from the *portpc* subdirectory to the main *tpa* directory and the *make* command executed again.

Some Lahey compiler options that are used to create the code to run as expected are described in the next sections.

-winconsole

The *win* part of this option makes the program a native NT windows program. The *console* part assures that all screen output is directed to one DOS window and that the window does not get deleted on termination of the program.

- nvm The -nvm option assures the NT memory manager is used instead of the Lahey virtual memory manager, which is a DOS-based memory manager.
- dbl The -dbl option is used to compile *snllhs.f* and assures the numeric outputs are the same under NT as those obtained under UNIX. The code compiled under NT without -dbl gives better precision (i.e., more decimal places) than the code compiled under UNIX without the -r8 option. The output is precise under NT while imprecise under UNIX.

The -dbl option is also used to compile *exec.f* and its subprograms so that all real variables, functions, and constants are promoted to double precision. This eliminates many problems where parts of the TPA code are not double precision and gives output that compares closely to UNIX. An important equation in the subprogram *uzft.f* is: $SAT(I) = (I - 5.) / 100$. This equation is implicit single precision on the right hand side and double precision on the left. The Lahey FORTRAN compiler, which compiles according to the FORTRAN standard, implicitly converts the single precision right hand expression to a double precision number, which is only correct to 6 decimal places (single precision) and not 15 (double precision) as expected. The explicit double precision FORTRAN statement that executes as expected is $SAT(I) = (DBLE(i) - 5.) / 100$. or $SAT(i) = (i - 5.D0) / 100$. TPA Version 3.2 code is compiled with the less precise, implicit statement. This part of the code has not been modified so that the TPA Version 3.2 and TPA3.2PC β codes will give the same answers when compiled under UNIX. Now that the NT code is compiled -dbl, the implicit single precision right hand side is promoted correctly under NT. This changes the answers somewhat from those obtained with the same code compiled under UNIX. To get the same answers from TPA3.2PC β under NT and UNIX, the equation must be changed to the explicit double precision version or the code must be compiled with Sun FORTRAN Version 4.2 using the compile option:

```
-xtypemap=real:64,double:64,integer:mixed
```

which, according to the FORTRAN User's Guide (Sun Microsystems, Inc., 1996), is equivalent to using the -dbl Lahey FORTRAN option under NT on the PC.

The size of the executables can be made significantly smaller without affecting the output values by modifying certain parameters controlling array sizes. These parameters are shown in table A-1 in order of effect on the executable program size.

Compiling *nefmks.f* without the -dbl option will reduce the executable to half its former size. It is unknown how great an effect this will have on the final TPA results.

A.1.1 The Lahey Fortran 90 Make Utility

The Lahey FORTRAN compiler uses proprietary *make* utilities for compilation and linking. These utilities can be run from the DOS command line or from the Lahey editor, ED. For simplicity, it is assumed that all commands shown in this section will be executed from a DOS command line. If they are to be executed from the Lahey editor, then the Lahey documentation should be consulted.

Table A-1. Key parameters impacting program size

| Variables | Files | Current Values | Needed to Run Basecase |
|------------------------|-------------------|----------------|------------------------|
| <i>maxrealizations</i> | <i>exec.f</i> | 500 | 1 (300 is reasonable) |
| <i>maxnnucl</i> | <i>maxnnucl.i</i> | 43 | 20 |
| <i>maxnsubarea</i> | <i>maxnsuba.i</i> | 14 | 7 |

The Lahey make utility is called AUTOMAKE and is executed from the DOS command line using the command of the same name (*automake*).

AUTOMAKE uses configuration files to control the compilation of large programs. These configuration files are denoted by a *.fig* extension. The exact format of these files is outlined in the Lahey documentation. Essentially, they consist of three parts: (i) compiler options, (ii) files to compile, and (iii) linker options. The order of these parts in the configuration file is important. This point is unclear in the Lahey documentation, but is critical nonetheless. The **.fig* files included with the TPA code should be sufficient without modification and provide good examples of **.fig* files.

The AUTOMAKE utility, by default, uses the file *automake.fig* in the current directory for configuration information. If one does not exist, AUTOMAKE will use a default configuration. To request that another file be used instead, the *-FIG filename* command line option should be used. For example:
 c:\cnwra\tpa\automake -FIG myautomake.fig

The AUTOMAKE utility produces a number of additional files when run:

**.dep* files store dependency information of the source code.

Note: All *.dep* files in the TPA code directory and subdirectories must be removed before the codes first compile. After that, they should not be deleted.

**.bat* files contain command line compile and link instructions. These are executed in a batch file when *automake* is run. They can be removed at will. Not all **.bat* files are necessarily created by AUTOMAKE. An example is the file *make.bat*.

A.1.2 Creating an executable

The TPA code is compiled and linked using the Lahey *make* utility. Configuration and batch files have been provided to facilitate this process. The necessary files are identified in table A-2.

The compilation is currently configured for an Intel Pentium class machine running NT or Windows 95. The Lahey documentation should be consulted to change this configuration.

Table A-2. Configuration and batch files for compiling and linking the TPA code using the Lahey utility

| Batch file | Configuration File | Description |
|---------------------------|---------------------------|--|
| <i>make.bat</i> | — | compiles and links the main executable and calls <i>makeall.bat</i> in subdirectory <i>codes</i> |
| | <i>automake.fig</i> | controls the compile and link of the main executable |
| <i>codes\makeall.bat</i> | — | compiles and links all standalone programs |
| | <i>codes\ashplume.fig</i> | controls the compile and link of the <i>ashplume.exe</i> standalone code |
| | <i>codes\ebfilt.fig</i> | controls the compile and link of the <i>ebfilt.exe</i> standalone code |
| | <i>codes\failt.fig</i> | controls the compile and link of the <i>failt.exe</i> standalone code |
| | <i>codes\nefmks.fig</i> | controls the compile and link of the <i>nefmks.exe</i> standalone code |
| | <i>codes\releaset.fig</i> | controls the compile and link of the <i>releaset.exe</i> standalone code |
| | <i>codes\snllhs.fig</i> | controls the compile and link of the <i>snllhs.exe</i> standalone code |
| <i>ccdf\maketccdf.bat</i> | <i>ccdf\automake.fig</i> | controls the compile and link of the <i>tccdf.exe</i> standalone code |

The commands used in the *make.bat* batch file perform the following logic:

Check if *lf90* exists. If *lf90* exists, continue. If it does not, then change the path environment variable and try again. If it is still not found, warn the user and quit.

```
lf90 > NUL
if errorlevel 1000 echo lf90 not found.
if errorlevel 1000 set path=%path%;c:\lf9045\bin
lf90 > NUL
if errorlevel 1000 echo lf90 compiler still not found.
if errorlevel 1000 goto QUIT
```


Compile the standalone and *ccdf* codes:

```
cd codes
makeall
cd ..\ccdf
automake
cd ..
```

Compile the main executable:

```
automake
```

The previous commands are included in the main TPA directory in the batch file, *make.bat*, which contains the commands to compile and link TPA3.2PC β .

A.2 PROGRAM EXECUTION/ERRORS

The Lahey compiler is designed to compile under DOS, Windows 3.1, Windows 95, and NT. The way the TPA code is currently compiled, it should run on both Windows 95 and NT, however, it has only been extensively tested for NT. There was one test run on Windows 95, as described in the next section. There is not enough memory space available to compile the code under DOS and therefore, that option has not been included in the make files. Compiling to run under Windows 3.1 is considered dubious and has also not been included.

The test run under Windows 95 was performed with the precompiled version on a system with

- 133 MHz Intel Pentium
- 98 MB physical memory
- 265 MB hard disk space after Windows 95 startup plus 192 MB of free virtual memory disk space available. Since TPA3.2PC β and virtual memory disk space were on the same disk, there was 495 MB available before the installation and execution of TPA3.2PC β and 5 MB left after.

The basecase run (1 realization) was successfully completed.

Adequate disk space is crucial for compiling and executing the code under NT as well as under UNIX. Otherwise, untraceable errors may occur. If only a few hundred megabytes are available, the disk may quickly and unexpectedly fill up. Under NT, the executable, *nefmks.exe*, uses 74 MB of disk space as compared to .5 MB under UNIX. Other NT executables in the *codes* subdirectory also use comparatively larger amounts of disk space. Each of these executable files is copied to the *run* directory when TPA is executed, thus using 120 MB of disk space before TPA starts its calculations. The amount of disk space needed can be determined from table 2-1 in conjunction with the guidelines in the TPA Version 3.2 user's guide (Mohanty and McCartin, 1998). Disk space should always be checked before each compile or run. A DOS *dir* command shows the remaining amount of free space left on the disk at the end of its output to the screen.

The virtual memory *minimum* should be set at 256 MB at the least. The recommended value is 400 MB *minimum* and *maximum* so that it will not have to be increased later by NT. The virtual memory/swap file should also all be on one disk, if feasible. If the program is run on a different disk than the swap disk, it will help increase speed somewhat since the disk accesses will not be competing. If the virtual memory disk space is on the same disk as TPA3.2PCβ and the virtual memory has been set by the system administrator such that the *minimum* is less than the *maximum*, it means the *maximum* minus the *minimum* extra megabytes of disk space may be used by the operating system and may not be available for TPA3.2PCβ files when the program needs to write them. This can cause an *insufficient disk space* error as both virtual memory and TPA3.2PCβ vie for the same disk space and use all of it. While the operating system virtual memory management system can usually perform with less than the *maximum* specified by the user, TPA3.2PCβ must have disk space available for writing its data files or it will abort.

When executing TPA under NT, most other applications currently running will greatly slow down. If memory is taken up by many opened (even if not used) processes, then more disk virtual memory accesses take place and programs start to operate slowly. Even when the TPA code is executing under the best circumstances, the disk is constantly being accessed. Many manufacturers provide a bus master controller (Direct Memory Access disk controller) driver that can be installed to increase disk access speed (Gardinier, 1998).

TPA3.2PCβ would usually be executed from a DOS window. But since it is compiled as an NT executable, it can also be executed by double-clicking a preestablished item on the desktop.

As mentioned under program compilation, if a run causes *out of handles* errors or causes the output from some standalone codes to be lost, then the TPA 3.2PCβ code has not been linked with the *lfsystem.obj* file (Lahey fix to lf90v4.5e).

Most errors occur as a result of the *TPA_DATA* and *TPA_TEST* environment variables not being set properly. *TPA_TEST* must provide the name of the directory that is the parent directory of the *codes* subdirectory. As an example, if the *codes* subdirectory is *d:\cnwra\tpa\codes*, then *TPA_TEST* must be *d:\cnwra\tpa*. *TPA_DATA* must provide the name of the directory that is the parent directory of the *data* subdirectory. Neither directory can contain a trailing slash (/) or backslash (\). This is automatically supplied by the TPA code. No parsing of the environment variables themselves is done. Thus, for NT, \ must be used for directory names, not the UNIX standard /. NT directory names are not case sensitive like UNIX directory names. Relative path names of all sorts may be used. For example, *TPA_TEST="."* will work so long as the code is run from the parent directory of the *codes* subdirectory. The *TPA_DATA* and *TPA_TEST* variables can be checked at any time by typing the command: *set*.

An error that occurs only when input files are improperly moved between UNIX and NT is

```
***>>> Error in Reader <<<***
need to use keyword: endoffile as last line to be read in the tpa.inp file
```

The solution to this problem is stated in section A.4.2.

All standard errors explained in the TPA Version 3.2 user's guide also apply to TPA3.2PCβ.

A.3 NT/DOS FILENAMES

NT supports free format filenames (long filenames), which can even include spaces. DOS uses fixed format filenames composed of at most eight characters plus an optional extension that is composed of a period followed by at most three characters (8.3 format). A DOS window under NT is actually an NT Command Prompt (extended DOS emulator) window. This window supports long filenames. If the mode of the window changes, it may present the internally converted, DOS fixed format, version of the filename. Because the 8.3 format is accepted by most operating systems, all TPA3.2PC β FORTRAN source and output filenames adhere to the 8.3 format. Modifications to TPA3.2PC β to achieve compliance with the 8.3 format are shown in table A-3.

A.4 PORTABILITY ISSUES (UNIX VS NT)

To maintain TPA code portability, several issues must be addressed. These include adhering to specific coding procedures as well as considering operating system and FORTRAN implementation differences. Such issues are addressed in the next sections.

A.4.1 Writes and Prints to the Screen

A.4.1.1 Operating System Screen Messages

The TPA code contains a number of Operating System (OS) calls that spawn system processes. Under SunOS, most of these calls do not produce any output to the screen unless an error occurs. Under DOS, the equivalent commands often produce screen output. This output interferes with the normal TPA code output.

This behavior has been corrected by redirecting this extraneous output to the DOS *NUL* device. No action is taken under SunOS. The decision to dump output to the *NUL* device under DOS is made by the subroutine *zportparseunixcmdtodos* called from the subroutine *zportsh*. The redirection to *NUL* is made if the command line does not already contain a redirection message, signaled by the presence of the > character. If this character is on the command line, the output will NOT be redirected to *NUL*. Certain types of screen output cannot be redirected by DOS. When these messages are produced, they appear on the screen and interfere with the TPA code output regardless of the > *NUL* redirection. Operating system messages are generally fatal. If such a message is displayed, the TPA code usually aborts on its own shortly thereafter. Some OS messages will not be fatal but will be printed to the screen while the code continues to run normally. These messages cannot be redirected under NT using the > *NUL* redirection technique.

A.4.1.2 Carriage Control

There are differences between the Lahey FORTRAN 90 compiler and the Sun FORTRAN 77 compiler regarding carriage control. The Lahey compiler strictly adheres to the FORTRAN standards for carriage control. The Sun compiler ignores carriage control and treats these as normal characters for output. To achieve carriage control on a Sun platform, the program output must be piped through the program *asa*.

Currently, the TPA code is written to use the Sun compiler behavior, which cannot be changed. When compiled under NT, this results in the first character of all formatted write and print statements treated

Table A-3. Revised file names in the scripts, supporting programs, and pre/post-processors

| From File Name | To File Name | Found In |
|-------------------------|---------------------|---------------------------|
| <i>relgwgs_c.res</i> | <i>rlgwgs_c.res</i> | <i>exec.f</i> |
| <i>dry_thick.dat</i> | <i>drythick.dat</i> | <i>nfenv.f</i> |
| <i>chloridemf.dat</i> | <i>chlrdmf.dat</i> | <i>fault.f</i> |
| <i>airpkdos_c.res</i> | <i>arpkds_c.res</i> | <i>exec.f</i> |
| <i>gwpkdos_c.res</i> | <i>gwpkds_c.res</i> | <i>exec.f</i> |
| <i>npkdoset_c.res</i> | <i>npkdst_c.res</i> | <i>exec.f</i> |
| <i>relccdf_c.res</i> | <i>rlccdf_c.res</i> | <i>exec.f</i> |
| <i>totdose_c.res</i> | <i>totdos_c.res</i> | <i>exec.f</i> |
| <i>cumrelease.out</i> | <i>cumrelse.res</i> | <i>codes/releaset.f</i> |
| <i>dignostic.out</i> | <i>diagnose.out</i> | <i>codes/releaset.f</i> |
| <i>echo_release.out</i> | <i>echo_rel.out</i> | <i>codes/releaset.f</i> |
| <i>frac_relrte.out</i> | <i>frac_rel.out</i> | <i>codes/releaset.f</i> |
| <i>echo_fail.dat</i> | <i>echofail.dat</i> | <i>codes/fault.f</i> |
| <i>treleasel.out</i> | <i>trelease.out</i> | <i>codes/releaset.f</i> |
| <i>peakfinder.f</i> | <i>peakfind.f</i> | <i>main tpa directory</i> |

as a carriage control character. This generally means the first character of every line is deleted since alphabetic characters are treated as spaces. Occasionally it may mean a line is not printed to the screen (i.e., lines beginning with a +), a line is skipped (i.e., lines beginning with a 0), or an entire page skipped (i.e., lines beginning with a 1).

There are three options available affecting the screen output under UNIX and NT: (i) the code is not changed, which allows the first character to be interpreted (usually deleted) under NT; (ii) the code is modified to provide for carriage control, and the output is piped through *asa* under SunOS, which achieves identical carriage control under NT; and (iii) a space is inserted at the beginning of every line (for carriage control), but the output is not piped through *asa* under SunOS. This last option allows the outputs to differ by a single space that is present under UNIX but not present under NT. None of these options has been tested for the TPA code.

The following should be noted with regard to carriage control. According to the FORTRAN standard, the default format (*) treats carriage control specially. The format automatically inserts a *space* carriage

control character (go to next line) at the beginning of the output line. Thus, no carriage control format characters are needed for these lines. For the default format (*), the Sun compiler and the Lahey compiler appear to both follow the FORTRAN standard.

Carriage control only affects output to the screen, not output to a file. Some modules may use output redirection to write files via the OS (i.e., `snllhs.e <tpa_lhs.lgs >lhse.out`). The most likely cause of problems created by this usage of redirection is if a subsequent module uses a file created through redirection and relies on the exact spacing of fields or symbols in that file. This problem is not occurring in the present version of the code.

A.4.2 Moving Files from UNIX to NT

Any editor can be used on the files in either UNIX or NT. Common NT editors are *EDIT* typed from a DOS prompt and *Notepad* selected with mouse from *Start:Programs:Accessories*. These are menu based, screen-type editors with arrow-key/mouse cursor control. Common Solaris/UNIX editors are *vi* typed from a UNIX prompt and *Textedit* selected with the mouse from third mouse button on the screen background: *Programs*. The editor, *vi*, is a terminal based editor (i.e., keyboard control only). *Textedit* is a screen-type editor similar to *Notepad*. A 32bit *vi* editor for NT has been included in the *portpc* subdirectory.

DOS/NT files end with <CR><LF>, UNIX files end with <LF>. These file endings for one OS cannot be read properly by the other. One of the following approaches can be used to avoid this problem.

- Use *ftp* to move files between the PC and UNIX computers. This converts automatically.
 - Example 1: If the TPA code is installed on a UNIX platform, use *ftp* to move all the files to the PC and the files will be ready to go.
 - Example 2: If TPA is run on the PC and the output is to be compared with a run under UNIX, use *ftp* to move the output files and the files will be converted.
 - Example 3: If a new *tpa.inp* is made and put on an anonymous *ftp* server, use *ftp* from either a UNIX computer or PC and get it. The new *tpa.inp* is ready to go (*ftp* adds/deletes the ^M as needed).
- Use *unix2dos* or *dos2unix* programs (under UNIX) to convert the source codes from one format to the other.
 - Example: *tpa.inp* has been edited on a UNIX platform. Convert it to DOS format with `unix2dos tpa.inp tpa.dos`. It can now be used under NT.
- Use an editor such as *vi* to add/delete the carriage return (^M).
 - Example 1: *tpa.inp* has been edited on a UNIX machine and saved on a diskette. Copy it from the diskette to a PC and convert it by editing with *vi* and using: `1,$s/$/ctrl-Vctrl-M/` so that ^M is added to the end of each line.

Example 2: *tpa.inp* is edited on a PC and copied to a shared NFS mounted hard disk. Someone else logs on to a UNIX platform and edits it with *vi* and uses `1, $s/ctrl-Vctrl-M//` so that `^M` are deleted. It can now be used on the UNIX platform.

A.4.3 Differences in UNIX and NT Output

The standalone codes, except for *snllhs.f* as currently compiled, exhibit no differences in precision to 12 decimal places between the NT and UNIX compiled versions. While the calculated distributions are correct under UNIX compiled with the current `-O4` optimization option, the actual values output are different from the other optimization levels (`-O3`, `-O2`, `-O1`, `-O0` give identical output). The output from *snllhs.f*, when compiled under NT, can be compared only to output when compiled with the `-O3` option or less under UNIX. As previously stated, compiling *snllhs.f* without the `-r8` option under UNIX gives imprecise output, so, the `-r8` option also must be used if the output under UNIX is to be compared to the output under NT. Compiling with the options `-O3` and `-r8` under UNIX and `-O3` and `-dbl` under NT gives identical output. The output is formatted to only seven decimal places. Since the actual numerical output is identical to 12 decimal places, the output files are identical. To get these results, the UNIX make file, *codes/Makefile*, must be edited and the `-O4` option for *snllhs.f* changed to `-O3 -r8`.

There exist small differences, at about the ninth decimal place, in the output of the TPA code. This is due to *uzflow.f* and other TPA subprograms and probably has been mostly fixed by compiling the NT code with `-dbl`. Even though the numbers match, the output from *uzflow* is not correct past eight places because many constants with limited accuracy have been embedded in the code. The UNIX code was compiled with FORTRAN 77 Version 3, which cannot be exactly compared with the PC code compiled with the `-dbl` switch. FORTRAN 77 Version 4.2 has a switch:

```
-xtypemap=real:64,double:64,integer:mixed
```

which should give identical outputs. This has not been tested.

As it stands, the outputs are identical in most cases and similar in the rest. The free format files cannot be directly compared (`diff` on UNIX, `fc` on PC) because the spaces and scientific notation of the format are expressed differently by the Sun and Lahey compilers. There is a special case of differences in the rounding of numbers written in formatted output. A number such as 2.45 may be written as 2.5 with Sun FORTRAN 77 and 2.4 with Lahey *lf90*. This is important only when the number is in a formatted file that is used as an input to another routine.

Sun and Lahey compilers handle single and double precision (real) variables and constants differently. The Lahey compiler adheres strictly to the FORTRAN standard, while the Sun compiler does not. Under certain conditions, this causes calculation differences that are a result of using mixed precision values in expressions. The pertinent conditions are shown in the following examples.

- Sun/Solaris:

The following three examples demonstrate the consequence of not using `-r8` option on a Sun/Solaris system:

Example 1: The code lines

```
r1 = 1.123456789
r2 = 1.123456789d0
print*, r1
print*, r2
```

will result in `r1= 1.123456789` and `r2= 1.123456789`.

Example 2: The code lines

```
REAL*8 xxx, z1, z2
xxx = 1.123456789
z1 = xxx/1.0
z2 = xxx/0.1
print*, z1
print*, z2
```

will result in `z1 = 1.123456789` and `z2 = 11.23456789`. This use of constants will give correct answers.

Example 3: The code lines

```
REAL*8 xxx, z1, z2
REAL*4 yyy1, yyy2
xxx = 1.123456789
yyy1 = 1.0
yyy2 = 0.1
z1 = xxx/yyy1
z2 = xxx/yyy2
```

will result in `z1 = 1.123456789` and `z2 = 11.234567722`.

Thus, without the `-r8` option, the use of single precision values greater than or equal to 1.0 (i.e., `yyy1`) will give correct results. For less than 1.0 (i.e., `yyy2`), the result will be incorrect. This behavior is observed with all arithmetic operators (i.e., `*`, `/`, `+`, and `-`).

The following also should be noted with regard to the precision on a Sun/Solaris system:

- Compiling with `-r8` makes `REAL*4` a `REAL*8` and makes double precision a `REAL*16`. `-dbl` is the same as `-r8` plus `INTEGER*4` are promoted to `INTEGER*8` and used that way.
- `-xtypemap=REAL:64,double:64,integer:64` is the same as `-dbl` on the PC. Sun FORTRAN 77 Version 4.2 is needed for this equivalence to be valid.

- PC/Lahey:

The following observations can be made when the previous examples are repeated without the `-dbl` option in a PC/Lahey system. In example 1, `r1` is not equal to `r2`. In example 2,

z2 is incorrect. In example 3, the results are the same as for the Sun/Solaris. This observation applies to all arithmetic operations (i.e., *, /, +, and -). It should be noted that compiling with `-dbl` makes `REAL*4` a `REAL*8` and makes `REAL*8` a `REAL*8`. This corrects the problems noted because all constants are promoted too.

In the FORTRAN standard, the first character output with a write statement is used for carriage control. The standard is not used on the Sun. Lahey uses it only for output going to the screen. Many files in TPA3.2PC β are captured from the screen output with redirections (`tpa.exe > tpa.out`). Therefore, this output will have the first character used for carriage control. There is no way to turn this *feature* off in the Lahey compiler. To get carriage control from Sun, output must be piped through the *asa* program. Thus, one can make a UNIX file look like a PC file using the command line:

```
cat snllhs.lse | asa > snllhs.lse.out
```

For a purely text file, the first character from each line can be removed. In *vi*, the command, `1,$s/^./`, will accomplish this.

A Sun FORTRAN function, *ieee_handler*, allows the TPA program to take action when an arithmetic error occurs. For example, *ieee_handler* can be set to stop the program when an error occurs such as division by zero. While the Lahey compiler has similar functions, the PC may give messages different from the Sun when such errors occur. The *ieee_handler* function is currently only implemented in *snllhs.f*. The TPA3.2PC β main program (*exec.f*) deletes all error flags just before it exits so if errors occurred, no error message is written and the run completes as if nothing happened.

A.4.4 CODE FRAGMENTS REQUIRING PORTING OF THE CODE

To port the TPA code from the original Sun Solaris OS to the target Microsoft Windows NT OS, Solaris and NT system dependant codes were replaced by system independent codes, which call all system dependant codes indirectly. For each supported OS platform, the system dependant code has been gathered into one module. The module for the desired platform will be compiled into the TPA code while the modules containing the OS dependent code for other platforms are ignored.

When modifying the TPA code, OS independent code fragments must be used instead of their Sun FORTRAN equivalents. The code fragments to be replaced and the fragment that replaces them are summarized in table A-4.

A more detailed description of the interfaces, and implementation of these code fragments, can be found in the source code files *zportunx.f* and *zportpc.f*. All the ported code fragments are designed to resemble the original Sun code as closely as possible.

Table A-4. Fragments in TPA3.2PC β independent of the operating system

| Sun Code to be Replaced | Code Function | Replacement Code |
|--|--|--|
| FUNCTION sh | Executes a shell command | FUNCTION zportsh |
| FUNCTION ieee_flags | Checks, sets, or clears exception flags | FUNCTION zportieee_flags |
| FUNCTION ieee_handler | Handles exceptions | FUNCTION zportieee_handler |
| FUNCTION time | Returns the time in seconds | FUNCTION zporttime |
| FUNCTION ctime | Returns the time as a string | FUNCTION zportctime |
| FUNCTION fdate | Returns a date string | FUNCTION zportfdate |
| FUNCTION d_erf | Double precision Gaussian Error Function | FUNCTION zportderf |
| NO EQUIVALENT* | Converts filenames to OS equivalent | SUBROUTINE zportparseunixfilenametodos |
| *There is no equivalent under Sun FORTRAN for the function zportparseunixfilenametodos. It is intended to convert Sun file and path names in I/O statements to the equivalent file syntax for the supported platforms. | | |

The following guidelines should be followed when modifying the code:

- Use external statements as intended by the FORTRAN standard. Lahey FORTRAN requires this.
- Avoid using nonstandard FORTRAN features of various compilers.
- Avoid file and path names that do not conform to the DOS 8.3 filename standards.

A.4.4.1 Shell Commands

One frequently used command in the TPA code is *sh(command)*. Because these are UNIX specific commands, an equivalent command must be established for NT. For example, a UNIX command for copying data from the *data* directory to the local directory under a different file name is

```
istatus=sh('cp data/xxx.dat yyy.dat')
```

The equivalent command in the ported code is represented by `istatus=zportsh(command)` where *zportsh* is a subroutine developed as a substitute for *sh*. Therefore, that command will be changed to

```
istatus=zportsh('cp data/xxx.dat yyy.dat')
```

The function must be typed and externalized in the declaration section of the corresponding code by including the following statements at the top of the program/subroutine/function

```
integer zportsh
external zportsh
```

Under NT, the function *zportsh* always returns the integer 0, regardless of the exit status of the shell command. This is because there is no way to probe this value in NT. Under UNIX, it returns the actual exit status.

A.4.4.2 IEEE Floating Point Exception Handling Functions

Another command used in the TPA code is

```
ieeestate=ieee_flags('set','exception','common',ieeout)
```

Because this is a Solaris specific command, an equivalent command must be established in NT. For example,

```
ieeestate=ieee_handler('set','common',SIGFPE_DEFAULT)
```

is a command to set up the error handler. The equivalent ported command for the PC version is

```
ieeestate=zportieee_handler('set','common',SIGFPE_DEFAULT)
```

The functions, however, must be typed and externalized in the declaration statements of the corresponding code (i.e., the following lines must be included at the top of the program/subroutine/function):

```
integer zportieee_flags
external zportieee_flags
and/or
integer zportieee_handler
external zportieee_handler
```

In addition, if any of the UNIX *ieee handler* functions which are predefined (SIGFPE_DEFAULT, SIGFPE_IGNORE, or SIGFPE_ABORT) are used, they must also be externalized in the declaration statements of the corresponding code (i.e., the following lines are included at the top of the program/subroutine/function):

```
external SIGFPE_DEFAULT
and/or
external SIGFPE_IGNORE
and/or
external SIGFPE_ABORT
```

It should be noted that these functions may not behave exactly the same as the SunOS versions, since there are system-level differences in floating point and exception handling capabilities. Since Sun FORTRAN will not compile with external `SIGFPE_ABORT` because it is an intrinsic function, an explicit external function has been written using the same name. It uses the system routine `abort`, in the file `zportunx.f`.

A.4.4.3 File Names in Open Statements

For the command

```
open(UNIT=igetunitnumber('xxx'),FILE='codes/xxx',STATUS='UNKNOWN')
```

the equivalent ported code is

```
CHARACTER*30 tempstring
...
CALL ZPORTPARSEUNIXFILENAMETODOS('codes/xxx',tempstring)
open(UNIT=igetunitnumber('xxx'),FILE=tempstring,STATUS='UNKNOWN')
```

The following points should be noted:

- (i) Despite the name, under UNIX, the subroutine `ZPORTPARSEUNIXFILENAMETODOS` leaves the supplied *command* intact and unchanged.
- (ii) The temporary storage string (*tempstring*) must be at least as long as the constant command to parse (*codes/xxx*), and should be long enough to hold the equivalent DOS command. If the temporary string is not long enough, then the DOS command will be truncated and results will be undefined.
- (iii) The temporary string must be typed in the declaration statements of the corresponding code (i.e., the following lines should be included at the top of the program/subroutine/ function):

```
CHARACTER*## tempstring
```

where `##` is the length of the string (see ii), and *tempstring* is a unique identifier.

- (iv) The subroutine must be externalized in the declaration statements of the corresponding code (i.e., the following lines must be included at the top of the program/subroutine/ function):

```
external zportparseunixfilenametodos
```

A.4.4.4 Time and Date Functions and Subroutines

The original UNIX code is

```
INTEGER*4 thetimeis
CHARACTER*24 timestring, datestring, datestringtoo
...
thetimeis=time()
timestring=ctime(thetimeis)
datestring=fdate()
call fdate(datestringtoo)
```

The equivalent ported code is

```
INTEGER*4 thetimeis
CHARACTER*24 timestring, datestring, datestringtoo
...
thetimeis=zporttime()
timestring=zportctime(thetimeis)
datestring=zportfdatefun()
call zportfdate(datestringtoo)
```

The following points should be noted:

- (i) The functions must be typed and externalized in the declaration statements of the corresponding code (i.e., the following lines must be included at the top of the program/subroutine/function):

```
integer*4 zporttime
external zporttime
and/or
character*## zportctime
external zportctime
and/or
character*## zportfdatefun
external zportfdatefun
and/or
external zportfdate
```

where ## should be at least 24 (any more than 24 is unnecessary, and the string will be padded with blanks)

- (ii) These functions/subroutines should behave identically to the matching SunOS functions/subroutines.
- (iii) The behavior of the function *zportctime* is unique and described in the file *zportpc.f*.

A.4.4.5 Numerical Recipes Library Functions

There are several numerical recipes library functions used in the UNIX version of the TPA code. The only one specifically ported is

```
erfvalue=d_erf(xvalue)
```

The equivalent ported code is

```
erfvalue=zportderf(xvalue)
```

The functions must be typed and externalized in the declaration statements of the corresponding code (i.e., the following lines are included at the top of the program/subroutine/function):

```
DOUBLE PRECISION zportderf
external zportderf
```

A.4.4.6 Procedure Interfaces

The ported function and subroutine interfaces are given in the following list. All interfaces are contained in *zportpc.f* and *zportunx.f*. These files should be referred to for a complete reference to the interfaces described.

- DOUBLE PRECISION FUNCTION ZPORTDERF(X)
DOUBLE PRECISION X

— X is an IN only variable
- INTEGER FUNCTION ZPORTSH(COMMAND)
CHARACTER*(*) COMMAND

— *COMMAND* is an IN only variable

— *COMMAND* should be at least 2 characters and less than 256 characters
- INTEGER*4 FUNCTION ZPORTTIME()
- CHARACTER*(*) FUNCTION ZPORTCTIME(SYSTIME)
INTEGER*4 SYSTIME

— *SYSTIME* is an IN only variable

— The return value is no more than 24 characters
- INTEGER FUNCTION ZPORTIEEE_FLAGS(MYACTION,MODE,ININ,OUTOUT)
CHARACTER*(*) MYACTION
CHARACTER*(*) MODE
CHARACTER*(*) ININ
CHARACTER*(*) OUTOUT

- *MYACTION* is an IN only variable
- If *MYACTION* is a variable, it should be at least eight characters. Allowed values are
 - i. 'get'
 - ii. 'set'
 - iii. 'clear'
 - iv. 'clearall'
- *MODE* is an IN only variable
- If *MODE* is a variable, it should be at least nine characters. Allowed values are
 - i. 'direction'
 - ii. 'precision'
 - iii. 'exception'
- *ININ* is an IN only variable
- If *ININ* is a variable, it should be at least nine characters. Allowed values that depend on the value of *MODE* are
 - i. 'nearest'
 - ii. 'tozero'
 - iii. 'negative'
 - iv. 'positive'
 - v. 'extended'
 - vi. 'double'
 - vii. 'single'
 - viii. 'inexact'
 - ix. 'division'
 - x. 'underflow'
 - xi. 'overflow'
 - xii. 'invalid'
 - xiii. 'all'
 - xiv. 'common'
- *OUTOUT* is an OUT only variable
- *OUTOUT* should be at least 13 characters. The possible values are
 - i. 'nearest'
 - ii. 'tozero'
 - iii. 'negative'
 - iv. 'positive'
 - v. 'extended'
 - vi. 'double'

- vii. 'single'
- viii. 'inexact'
- ix. 'division'
- x. 'underflow'
- xi. 'overflow'
- xii. 'invalid'
- xiii. 'all'
- xiv. 'common'
- xv. 'not available'
- xvi. '' (empty string)

- INTEGER FUNCTION ZPORTIEEE_HANDLER(MYACTION, EXCEPTION, HANDLER)
 CHARACTER*(*) MYACTION
 CHARACTER*(*) EXCEPTION
 INTEGER HANDLER
 EXTERNAL HANDLER

— *MYACTION* is an IN only variable

— If *MYACTION* is a variable, it should be at least eight characters. Allowed values are

- i. 'get'
- ii. 'set'
- iii. 'clear'
- iv. 'clearall'

— *EXCEPTION* is an IN only variable

— If *EXCEPTION* is a variable, it should be at least nine characters. Allowed values are

- i. 'inexact'
- ii. 'division'
- iii. 'underflow'
- iv. 'overflow'
- v. 'invalid'
- vi. 'all'
- vii. 'common'

— *HANDLER* is an integer function

- SUBROUTINE ZPORTPARSEUNIXCMDTODOS(COMMAND, NEWCMND)
 CHARACTER*(*) COMMAND
 CHARACTER*(*) NEWCMND

— *COMMAND* is an IN only variable

— *COMMAND* should be at least 2 characters and less than 256 characters

— *NEWCMND* is an OUT only variable

- *NEWCMND* should be at least 10 characters greater than *COMMAND*
- SUBROUTINE ZPORTPARSEUNIXFILENAME(TODOS (COMMAND, NEWCMND)
CHARACTER* (*) COMMAND
CHARACTER* (*) NEWCMND
 - *COMMAND* is an IN only variable
 - *COMMAND* should be at least 2 characters and less than 256 characters
 - *NEWCMND* is an OUT only variable
 - *NEWCMND* should be equal to or greater in size than *COMMAND*

A.5 REFERENCES

- Gardinier, K. 1998. *Windows NT Performance Tuning & Optimization*. Berkeley, CA: McGraw-Hill.
- Mohanty, S., and T.J. McCartin. 1998. *Total-system Performance Assessment (TPA) Version 3.2 Code: Module Descriptions and User's Guide*. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.
- Sun Microsystems. 1996. *Fortran User's Guide: FORTRAN 77 Compiler 4.2, Fortran 90 Compiler 1.2*. Mountain View, CA: Sun Microsystems, Inc.

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE: TPA Version 3.2 Personal Computer (PC) Beta

December 1, 1998

Total-System Performance Assessment (Scientific and Engineering Software) Version 3.2 Personal Computer (PC) Beta

NOTE: This version of the TPA Software contains the modification of TPA Version 3.2 to enable its operation on a personal computer. An electronic scientific notebook assigned to Sitakanta Mohanty has been utilized as the change documentation method.

1. **This Design Verification Report is prepared by:** Bruce Mabrito in conjunction with Michael Muller.
Full Title of CNWRA scientific and engineering software: Total-System Performance Assessment (TPA) Version 3.2 Personal Computer (PC) Beta.
Demonstration work station: Pentium 200 mHz PC (named "Phoenix") in CNWRA office A204.
Operating System: Windows NT 4.0

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

NOTE: A very straightforward and short SRD was prepared by M. Muller (of SwRI Division 15) and was approved after-the-fact by the CNWRA PA Element Manager.

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

NOTE: A very straightforward SDP was prepared by M. Muller and was approved after-the-fact by the CNWRA PA Element Manager.

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

NOTE: An electronic scientific notebook (No. 170) was utilized and contains module level documentation.

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

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: The latest CNWRA/SwRI Contract No. (NRC-02-97-009) was not reflected in the source code module header of TPA Version 3.2 PC Beta. The software developer was made aware of this but it was agreed not to change it at this late date in the development of the code. Other requirements were fulfilled.

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: The TPA SwRI Div. 15 Version 3.2 PC Beta Procedure for Installing and Modifying Total-System Performance Assessment (TPA) Version 3.2 PC Beta Code (dated December 1998) was available during the Design Verification activities. A separate CNWRA TPA V 3.2 PC Beta handout will be written by the CNWRA and sent to the NRC later.

Are there basic instructions for the use of the software?

☒ YES NO N/A

NOTE: The extensive User's Manual is that published in September 1998 and relates directly to the primary TPA Version 3.2 Code (that is, the TPA Version 3.2 Code Module Description and User's Guide).

10. Acceptance Testing

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

☒ YES NO N/A

NOTE: TPA V3.2 PC Beta was compiled, linked, executed and tested on a CNWRA PC.

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

YES

NO

N/A

NOTE: Acceptance testing on Sun platforms with the Solaris O.S. and on PCs with the Windows NT version 4.0 has been completed. Summaries are in the electronic scientific notebook. Additionally, a test results packet is included in this folder.

Have installation tests been performed on the target platform?

YES

NO

N/A

NOTE: On the PCs at the CNWRA.

11. Configuration Control

Is the Software Summary Form completed and signed?

YES

NO

N/A

If no, explain:

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

YES

NO

N/A

If no, explain: The technical description is given in the Software Requirements Description given in TPA 3.2.

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

YES

NO

N/A

NOTE: For the TPA V 3.2 PC Beta, the answer is yes.

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

YES

NO

N/A

Michael Muller 12/1/98
Dec 1, 1998
Michael Muller Date

Bruce Mabrito 12/1/98
12/1/98
Bruce Mabrito Date

CNWRA TPA Software Co-Developer

CNWRA Software Custodian

Attachments/

Original to: Software Folder

cc: CNWRA Software Developer/Cognizant EM/S. Mohanty

SOFTWARE REQUIREMENTS DESCRIPTION - TPA Version 3.2PC β **(Personal Computer)**

1.0 SOFTWARE FUNCTION

The TPA Version 3.2 code was developed for UNIX Solaris workstations. The TPA Version 3.2 code will be modified so that the same source code runs on a UNIX Solaris workstation and on a PC with Windows NT using Lahey FORTRAN 90.

2.0 TECHNICAL BASIS: PHYSICAL AND MATHEMATICAL MODEL

(not applicable - the same as TPA Version 3.2 code)

3.0 COMPUTATIONAL APPROACH

3.1 Data Flow and User Interface

(not applicable - the same as TPA Version 3.2 code)

3.2 Hardware and Software Requirements

- **Target platform**
SUN and PC
- **Operating system**
Solaris and Windows NT
- **Programming language**
FORTRAN 77 (UNIX) and Lahey FORTRAN 90 (Windows NT)

3.3 Graphics Requirements

(not applicable - the same as TPA Version 3.2 code)

3.4 Pre- and Post-Processors

(not applicable - the same as TPA Version 3.2 code)

4.0 REFERENCES

(not applicable - the same as TPA Version 3.2 code)

APPENDICES - (not applicable)

Approved: Gordon Wilkerson 12/1/98

SOFTWARE DEVELOPMENT PLAN

SOFTWARE DEVELOPMENT PLAN - TPA Version 3.2PC β (Personal Computer)

1.0 SCOPE

The TPA Version 3.2PC β code will allow users to run the same source code on Sun and PC operating systems. The current version of the code (TPA Version 3.2) was developed using FORTRAN 77 on the Sun Solaris operating system. The PC version of the TPA code will utilize the PENTIUM/NT operating system with Lahey FORTRAN 90. This Software Development Plan describes the approach for producing a version of the TPA code that will run on both the Sun and PC operating systems.

2.0 BASELINE ITEMS

The software product generated in this project will be FORTRAN source code that runs the TPA code on UNIX and NT systems. The same FORTRAN source code will be compiled with two different *makefiles* using FORTRAN 77 on the Sun Solaris operating system and using Lahey FORTRAN 90 on the PENTIUM/NT operating system. A TPA Version 3.2 PC β user's guide will describe the system requirements and differences between PC version and the TPA Version 3.2 code. The TPA Version 3.2 code will be modified to develop TPA Version 3.2 PC β .

3.0 PROJECT MANAGEMENT

3.1 Work Breakdown Structure

There are two general tasks that will be performed in producing TPA Version 3.2 PC β code. First, the TPA Version 3.2 code will be modified to generate an executable on the PC NT system while preserving the functionality on the Sun UNIX system. Following the completion of coding changes, the TPA Version 3.2 PC β code will be tested on both the Sun and NT systems and the results benchmarked against output from the TPA Version 3.2 code.

3.2 Projected Schedule

The estimated level of effort to complete the development and testing of the TPA Version 3.2 PC β code is 15 weeks (600 hours). The effort will involve producing a single source code that can be maintained for both the Sun and PENTIUM/NT systems.

3.3 Staffing

The staffing requirements for developing and testing the TPA Version 3.2 PC β code include one SwRI staff programmer working half-time or a student programmer working full-time. The testing will be performed by one SwRI staff member working part-time.

3.4 Risk Management

The major risks associated with developing the TPA Version 3.2 PC β code include evaluating whether: Sun FORTRAN 77 and Lahey FORTRAN 90, specifically non-standard FORTRAN features, are compatible; TPA results would be the same on Sun and NT systems; PENTIUM/NT systems could

run the TPA code; the PC execution time is comparable to the Sun; and the PC system requirements are feasible. Any of these four risks could have significant impacts on the project by either stopping the development of the TPA Version 3.2 PC β code or forcing a change in the planned approach, such the operating system (i.e., NT), the FORTRAN compiler (i.e., Lahey FORTRAN 90), and the system requirements (i.e, disk space, memory, and processor). For the risks associated with cost, schedule, resources, and functionality identified previously, the following table provides estimates of the risk probability and impact, and options for dealing with the identified risks.

| Risk | Probability | Impact of Risk | Option |
|--|--|--|--|
| Sun and Lahey FORTRAN compatibility | MEDIUM (based on some non-standard FORTRAN calls in the TPA Version 3.2 code) | Potential impact is to necessitate a change in the TPA Version 3.2 code | Modify the TPA Version 3.2 code |
| Obtaining the same TPA results on Sun and NT systems | HIGH (based on no previous comparison) | Potential impact is to stop the development of the TPA Version 3.2 PC β code | Modify the TPA Version 3.2 code or regard the differences as acceptable at a given level of significance |
| NT system compatibility | HIGH (based on the lack of previous information in porting from Sun to PC systems) | Potential impact is to stop the development of the TPA Version 3.2 PC β code | Select another PC operating system, such as DOS, WINDOWS95, Solaris, or LINUX. |
| PC system resource requirements | HIGH (based on significant Sun system requirements being) | Potential impact is to stop the development of the TPA Version 3.2 PC β code | Upgrade the PC system or decrease the system requirements of the TPA Version 3.2 PC β code, such as array maximum dimensions which will decrease memory requirements |

4.0 DEVELOPMENT PROCEDURES

4.1 Hardware and Software Resources

The anticipated hardware resources in this project are a PC PENTIUM system with NT having at least a 200 MHz processor, 2.0 GB of available disk space, and 64 MB of memory. Additionally, the Ultra 2 system ("bigbend") will be needed to test the UNIX version of the TPA Version 3.2 PC β code. The anticipated software requirements include NT, Lahey FORTRAN 90, UNIX, Sun FORTRAN 77, and the TPA Version 3.2 code.

4.2 Software Development Lifecycle

The TPA Version 3.2 PC β code will be developed from the TPA Version 3.2 code and testing will be conducted to compare output from the UNIX and NT codes with the TPA Version 3.2 code. Once testing and the evaluation of the performance of the TPA Version 3.2 PC β code are completed the TPA Version 3.2 PC code will be distributed and maintained. The software development lifecycle is divided into five phases:

- Phase 1: identify all system dependent routines, write wrapper functions for the system dependent routines (wrapper functions are compiled for the specific target system), and test that the wrapper functions will compile and run without aborting on UNIX and PC systems
- Phase 2: Conduct testing to verify that results from the UNIX and PC implementation of the wrapper functions are consistent
- Phase 3: Using the TPA code from Phase 2, modify the TPA Version 3.2 code to develop the TPA Version 3.2 PC β code
- Phase 4: Test the TPA Version 3.2 PC β code from Phase 3
- Phase 5: Deliver and maintain the TPA Version 3.2 PC β code.

4.3 Coding

The TPA Version 3.2 PC β code will be developed in Lahey FORTRAN 90 using the same coding convention utilized in the TPA Version 3.2 code, such as modularization and internal documentation. The implementation will involve using two different FORTRAN *makefiles* that employ either Sun FORTRAN 77 or Lahey FORTRAN 90 features. In this way, the changes to the TPA Version 3.2 code source code will be minimized.

4.4 Acceptance Testing and Analysis

Acceptance testing and analysis will be performed after completing the TPA Version 3.2 code modifications. The results from test cases will be documented in scientific notebooks and in hardcopies, electronic files, and on tapes and disks.

5.0 CONFIGURATION MANAGEMENT PLAN (CMP)

5.1 Tools

There are no formal software configuration management tools that will be used in developing the TPA Version 3.2 PC β code.

5.2 Configuration Identification

After completing modifications to the TPA Version 3.2 code and testing the changes, the TPA Version 3.2 PC β code will be placed under configuration control as TPA Version 3.2 PC.

5.3 Configuration Procedures

The configuration procedures for the TPA Version 3.2 PC β code will be the same as those followed with the TPA Version 3.2 code. It is anticipated that there will be no changes needed in any of these configuration procedures.

6.0 REFERENCES

(none)

7.0 APPENDICES

(none)

Approved: Gordon Wittmeyer 12/1/98

postscript

JOB 58

D: pa unscreen.out

installation test

on clerie

*(bas. case
typing)*

d: /tpa/run

Creator: Windows NT 4.0

Creation Date: 10:11 11/2/1998

Submit queue: Ethernet

Submitted: 87:20:19

Started: 87:20:20



QMS 1725 Print System

QMS 1725 (1st floor)

```
=====
exec: Welcome to TPA Version 3.2PCbeta
Job started: Mon Nov 02 10:04:42 1998
=====
```

pecified Global Parameters:

```
Compliance Period = 10000.0 (yr)
Maximum Simulation Time = 10000.0 (yr)
Number Of Realizations = 1
Number Of Subareas = 7
Volcanism scenario = 0 (yes=1, no=0)
Faulting scenario = 0 (yes=1, no=0)
Seismic scenario = 1 (yes=1, no=0)
Distance to Receptor Group = 20.0 (km)
```

```
**>>> CAUTION: CHECKING OF NUCLIDES AND CHAINS IS DISABLED <<<**
**>>> You may not be using the standard chains specified <<<**
**>>> in the invent module. <<<**
**>>> (see "CheckNuclidesAndChains(yes=1,no=0)" in tpa.inp)<<<**
```

```
he specified path for data = $TPA_DATA/
he specified path for codes = $TPA_TEST/
```

*To modify global parameters or the path, stop code execution using control-C**

```
-----
subarea 1 of 7      realization 1 of 1
-----
```

xec: calling uzflow

xec: calling nfenv

xec: calling ebsfail

*** No Corrosion WP Failure ***

xec: calling seismo

xec: failed WPs from INITIAL event = 16 at time = 0.0 yr

*** failed WPs: 16 out of 1663 ***

xec: calling ebsrel

Highest release rates from Sub Area 1

```
Tc99 8.3873E-03 [Ci/yr/SA] at 2.540E+03 yr
Ni59 1.6417E-03 [Ci/yr/SA] at 2.540E+03 yr
C14 8.0455E-04 [Ci/yr/SA] at 2.540E+03 yr
Se79 2.8366E-04 [Ci/yr/SA] at 2.540E+03 yr
Cs135 2.6908E-04 [Ci/yr/SA] at 2.540E+03 yr
Am241 1.8366E-04 [Ci/yr/SA] at 2.540E+03 yr
```

xec: calling uzft

Highest release rates from UZ

```
Tc99 8.3491E-03 [Ci/yr/SA] at 2.602E+03 yr
Ni59 1.6337E-03 [Ci/yr/SA] at 2.602E+03 yr
Se79 2.8206E-04 [Ci/yr/SA] at 2.602E+03 yr
Cs135 2.6773E-04 [Ci/yr/SA] at 2.602E+03 yr
Am241 1.7138E-04 [Ci/yr/SA] at 2.602E+03 yr
Np237 1.6827E-04 [Ci/yr/SA] at 1.000E+04 yr
```

xec: calling szft

Highest release rates from SZ

```
Cm245 2.2786E-05 [Ci/yr/SA] at 8.691E+03 yr
I129 1.2110E-05 [Ci/yr/SA] at 1.000E+04 yr
Cl36 5.1927E-06 [Ci/yr/SA] at 1.000E+04 yr
Cm246 2.7135E-06 [Ci/yr/SA] at 8.101E+03 yr
Np237 1.7309E-10 [Ci/yr/SA] at 1.000E+04 yr
Am241 6.0024E-14 [Ci/yr/SA] at 9.543E+03 yr
-----
```

subarea 2 of 7 realization 1 of 1

xec: calling uzflow

xec: calling nfenv

xec: calling ebsfail

*** No Corrosion WP Failure ***

xec: failed WPs from INITIAL event = 17 at time = 0.0 yr

*** failed WPs: 17 out of 1767 ***

xec: calling ebsrel

There is no EBS release

xec: calling uzft

There is no UZ release

xec: calling szft

There is no SZ release

subarea 3 of 7 realization 1 of 1

xec: calling uzflow

xec: calling nfenv

xec: calling ebsfail

*** No Corrosion WP Failure ***

xec: failed WPs from INITIAL event = 8 at time = 0.0 yr

*** failed WPs: 8 out of 855 ***

xec: calling ebsrel

Highest release rates from Sub Area 3

Tc99 9.2848E-03 [Ci/yr/SA] at 9.769E+03 yr

Ni59 1.7539E-03 [Ci/yr/SA] at 9.769E+03 yr

C14 3.6029E-04 [Ci/yr/SA] at 9.543E+03 yr

Cs135 2.9074E-04 [Ci/yr/SA] at 9.769E+03 yr

Se79 2.8437E-04 [Ci/yr/SA] at 9.769E+03 yr

Np237 9.8061E-05 [Ci/yr/SA] at 1.000E+04 yr

xec: calling uzft

Highest release rates from UZ

Tc99 9.2733E-03 [Ci/yr/SA] at 9.769E+03 yr

Ni59 1.7517E-03 [Ci/yr/SA] at 9.769E+03 yr

Cs135 2.9040E-04 [Ci/yr/SA] at 9.769E+03 yr

Se79 2.8404E-04 [Ci/yr/SA] at 9.769E+03 yr

Np237 9.8059E-05 [Ci/yr/SA] at 1.000E+04 yr

Cm245 4.3605E-05 [Ci/yr/SA] at 9.769E+03 yr

xec: calling szft

Highest release rates from SZ

Cm245 2.8226E-09 [Ci/yr/SA] at 1.000E+04 yr

I129 8.8129E-10 [Ci/yr/SA] at 1.000E+04 yr

Cl36 6.9318E-10 [Ci/yr/SA] at 1.000E+04 yr

Cm246 2.9274E-10 [Ci/yr/SA] at 1.000E+04 yr

Np237 1.6109E-16 [Ci/yr/SA] at 1.000E+04 yr

Am241 1.0231E-18 [Ci/yr/SA] at 1.000E+04 yr

subarea 4 of 7 realization 1 of 1

xec: calling uzflow

xec: calling nfenv

xec: calling ebsfail

*** No Corrosion WP Failure ***

xec: failed WPs from INITIAL event = 5 at time = 0.0 yr

*** failed WPs: 5 out of 472 ***

xec: calling ebsrel

Highest release rates from Sub Area 4

Tc99 9.1174E-03 [Ci/yr/SA] at 1.723E+03 yr

Ni59 1.7872E-03 [Ci/yr/SA] at 1.723E+03 yr

C14 1.0205E-03 [Ci/yr/SA] at 1.723E+03 yr

Am241 7.3673E-04 [Ci/yr/SA] at 1.723E+03 yr
 Se79 3.2172E-04 [Ci/yr/SA] at 1.723E+03 yr
 Cs135 3.0260E-04 [Ci/yr/SA] at 1.723E+03 yr

xec: calling uzft

Highest release rates from UZ

Tc99 8.2456E-03 [Ci/yr/SA] at 1.766E+03 yr
 Ni59 1.6161E-03 [Ci/yr/SA] at 1.766E+03 yr
 Am241 6.6946E-04 [Ci/yr/SA] at 1.766E+03 yr
 Se79 2.9094E-04 [Ci/yr/SA] at 1.723E+03 yr
 Cs135 2.7367E-04 [Ci/yr/SA] at 1.723E+03 yr
 Np237 2.1460E-04 [Ci/yr/SA] at 1.766E+03 yr

xec: calling szft

Highest release rates from SZ

Tc99 9.7650E-05 [Ci/yr/SA] at 1.000E+04 yr
 Cm245 8.1937E-06 [Ci/yr/SA] at 3.549E+03 yr
 I129 2.8825E-06 [Ci/yr/SA] at 4.093E+03 yr
 Cm246 1.3297E-06 [Ci/yr/SA] at 3.549E+03 yr
 Cl36 1.3246E-06 [Ci/yr/SA] at 3.549E+03 yr
 Np237 7.8361E-11 [Ci/yr/SA] at 1.000E+04 yr

subarea 5 of 7 realization 1 of 1

xec: calling uzflow

xec: calling nfenv

xec: calling ebsfail

*** No Corrosion WP Failure ***

xec: failed WPs from INITIAL event = 6 at time = 0.0 yr

*** failed WPs: 6 out of 654 ***

xec: calling ebsrel

There is no EBS release

xec: calling uzft

There is no UZ release

xec: calling szft

There is no SZ release

subarea 6 of 7 realization 1 of 1

xec: calling uzflow

xec: calling nfenv

xec: calling ebsfail

*** No Corrosion WP Failure ***

xec: failed WPs from INITIAL event = 7 at time = 0.0 yr

*** failed WPs: 7 out of 738 ***

xec: calling ebsrel

There is no EBS release

xec: calling uzft

There is no UZ release

xec: calling szft

There is no SZ release

subarea 7 of 7 realization 1 of 1

xec: calling uzflow

xec: calling nfenv

xec: calling ebsfail

*** No Corrosion WP Failure ***

xec: failed WPs from INITIAL event = 3 at time = 0.0 yr

*** failed WPs: 3 out of 278 ***

xec: calling ebsrel

Highest release rates from Sub Area 7

Tc99 1.0809E-03 [Ci/yr/SA] at 3.549E+03 yr

Ni59 2.0964E-04 [Ci/yr/SA] at 3.549E+03 yr
 Cl14 9.8848E-05 [Ci/yr/SA] at 3.549E+03 yr
 Se79 3.7955E-05 [Ci/yr/SA] at 3.549E+03 yr
 Cs135 3.6383E-05 [Ci/yr/SA] at 3.549E+03 yr
 Np237 3.2403E-05 [Ci/yr/SA] at 1.000E+04 yr

xec: calling uzft

Highest release rates from UZ

Tc99 1.0712E-03 [Ci/yr/SA] at 3.635E+03 yr
 Ni59 2.0768E-04 [Ci/yr/SA] at 3.635E+03 yr
 Se79 3.7551E-05 [Ci/yr/SA] at 3.635E+03 yr
 Cs135 3.6027E-05 [Ci/yr/SA] at 3.635E+03 yr
 Np237 3.2403E-05 [Ci/yr/SA] at 1.000E+04 yr
 Am243 1.3633E-05 [Ci/yr/SA] at 1.000E+04 yr

xec: calling szft

Highest release rates from SZ

Tc99 5.0896E-05 [Ci/yr/SA] at 1.000E+04 yr
 Cm245 4.3805E-06 [Ci/yr/SA] at 6.560E+03 yr
 I129 2.0859E-06 [Ci/yr/SA] at 8.101E+03 yr
 Cl36 9.2960E-07 [Ci/yr/SA] at 7.551E+03 yr
 Cm246 5.9295E-07 [Ci/yr/SA] at 6.113E+03 yr
 Np237 2.6422E-11 [Ci/yr/SA] at 1.000E+04 yr

xec: calling dcagw

Highest annual dose GW pathway

Cm245 1.7074E-02 [mrem/yr] at 8.101E+03 yr
 Cm246 2.0989E-03 [mrem/yr] at 7.551E+03 yr
 I129 1.7803E-03 [mrem/yr] at 1.000E+04 yr
 Tc99 6.4596E-05 [mrem/yr] at 1.000E+04 yr
 Cl36 2.7012E-05 [mrem/yr] at 9.323E+03 yr
 Np237 2.2960E-07 [mrem/yr] at 1.000E+04 yr

At end of TPI, annual dose GW pathway

Cm245 1.5858E-02 [mrem/yr]
 I129 1.7803E-03 [mrem/yr]
 Cm246 1.7119E-03 [mrem/yr]
 Tc99 6.4596E-05 [mrem/yr]
 Cl36 2.6933E-05 [mrem/yr]
 Np237 2.2960E-07 [mrem/yr]
 sum 1.9442E-02 [mrem/yr]

xec: end realizations

xec: Run Successfully Completed

```

                                aa
c Program Name:      TPA - Total-System Performance Assessment Code
c File Name:        exec.f
c File Date:        10/27/1998
c Release Version:   3.2PCbeta
c
c Client Name:       USNRC
c                   U. S. Nuclear Regulatory Commission
c                   NRC Office of Nuclear Material Safety and Safeg
uards
c                   Division of Waste Management
c Contract Number:   NRC 02-93-005
c
c NRC Contact        Tim McCartin (301) 415-6681
c
c CNWRA Contact:     Sitakanta Mohanty (210) 522-5185
c                   Center for Nuclear Waste Regulatory Analyses
c                   San Antonio, Texas 78238-5166
c                   smohanty@swri.edu
c
c Revisions:
c                   3.1.1      includes SPCRs 101 through 205
c                   3.1.2      includes SPCRs 206 through 224
c                   3.1.3      includes SPCRs 225 through 227
c                   3.1.4      includes SPCRs 228 through 231
c                   3.2        includes SPCRs 232 through 252
c                   3.2PCbeta  port of 3.2 to PC running NT4
c
c Documentation:     Predecisional "Total-System Performance Assessm
ent
c                   (TPA) Version 3.2 Code: Module Description and
c                   User's Guide", Center for Nuclear Waste Regulat
ory
c                   Analyses
c NUREG-Series Designator: N/A
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
cc
c
c
c = = = = =
=
c
c                   D I S C L A I M E R
c
c = = = = =
=
c
c "This computer code/material was prepared 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

```

aa

c Government. Neither the developer(s) of the code nor any of their
 c sponsors 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 privately

c 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 the program (including but not limited to loss of
 c data or data being rendered inaccurate or losses sustained by third
 c parties or a failure of the program to operate with other programs),

c even if you have been advised of the possibility of such damages or
 c for any claim by any other party."

c
 c = = = = =
 c

c by S. Mohanty, R. Janetzke, R. Rice, A. Lozano
 c R. Manteufel (initial version)

c
 c=====

====
 program exec

tpa.out

DATE/Time

```
=====
=
      exec: Welcome to TPA Version 3.2PCbeta
      Job started: Tue Dec 01 12:35:27 1998
=====
```

Specified Global Parameters:

```
      Compliance Period = 10000.0 (yr)
Maximum Simulation Time = 10000.0 (yr)
      Number Of Realizations = 400
      Number Of Subareas = 7
      Volcanism scenario = 0 (yes=1, no=0)
      Faulting scenario = 0 (yes=1, no=0)
      Seismic scenario = 1 (yes=1, no=0)
Distance to Receptor Group = 20.0 (km)
```

```
**>>> CAUTION: CHECKING OF NUCLIDES AND CHAINS IS DISABLED <<<**
**>>> You may not be using the standard chains specified <<<**
**>>> in the invent module. <<<**
**>>> (see "CheckNuclidesAndChains(yes=1,no=0)" in tpa.inp)<<<**
```

```
The specified path for data = $TPA_DATA/
The specified path for codes = $TPA_TEST/
```

```
*To modify global parameters or the path, stop code execution using co
ntrol-C**
```

pcl5

JOB 128

screen.out

installation test
on bigbend
(base case
training)

/b sero/rrice/
installation test

For: bigbend!rrice
Date: Tue Nov 3 09:44:48 CST 1998
Submit queue: IF 1 / Ethernet / UHSW
Submitted: Mon Nov 02 12:01:07 1998
Started: Mon Nov 02 12:01:07 1998

=====

exec: Welcome to TPA Version 3.2

Job started: Tue Nov 3 09:31:39 1998

=====

Specified Global Parameters:

Compliance Period = 10000.0 (yr)
 Maximum Simulation Time = 10000.0 (yr)
 Number Of Realizations = 1
 Number Of Subareas = 7
 Volcanism scenario = 0 (yes=1, no=0)
 Faulting scenario = 0 (yes=1, no=0)
 Seismic scenario = 1 (yes=1, no=0)
 Distance to Receptor Group = 20.0 (km)

>>> CAUTION: CHECKING OF NUCLIDES AND CHAINS IS DISABLED <<<
 >>> You may not be using the standard chains specified <<<
 >>> in the invent module. <<<
 >>> (see "CheckNuclidesAndChains(yes=1,no=0)" in tpa.inp)<<<

The specified path for data = /bscr1/rrice/tpa3.2/

The specified path for codes = /bscr1/rrice/tpa3.2/

To modify global parameters or the path, stop code execution using control-C

subarea 1 of 7 realization 1 of 1

exec: calling uzflow

exec: calling nfenv

exec: calling ebsfail

*** No Corrosion WP Failure ***

exec: calling seismo

exec: failed WPs from INITIAL event = 16 at time = 0.0 yr

*** failed WPs: 16 out of 1663 ***

exec: calling ebsrel

Highest release rates from Sub Area 1

| | |
|-------|---------------------------------------|
| Tc99 | 8.3873E-03 [Ci/yr/SA] at 2.540E+03 yr |
| Ni59 | 1.6417E-03 [Ci/yr/SA] at 2.540E+03 yr |
| C14 | 8.0455E-04 [Ci/yr/SA] at 2.540E+03 yr |
| Se79 | 2.8366E-04 [Ci/yr/SA] at 2.540E+03 yr |
| Cs135 | 2.6908E-04 [Ci/yr/SA] at 2.540E+03 yr |
| Am241 | 1.8366E-04 [Ci/yr/SA] at 2.540E+03 yr |

exec: calling uzft

Highest release rates from UZ

| | |
|-------|---------------------------------------|
| Tc99 | 8.3491E-03 [Ci/yr/SA] at 2.602E+03 yr |
| Ni59 | 1.6337E-03 [Ci/yr/SA] at 2.602E+03 yr |
| Se79 | 2.8206E-04 [Ci/yr/SA] at 2.602E+03 yr |
| Cs135 | 2.6773E-04 [Ci/yr/SA] at 2.602E+03 yr |
| Am241 | 1.7138E-04 [Ci/yr/SA] at 2.602E+03 yr |
| Np237 | 1.6827E-04 [Ci/yr/SA] at 1.000E+04 yr |

exec: calling szft

Highest release rates from SZ

| | |
|-------|---------------------------------------|
| Cm245 | 2.2786E-05 [Ci/yr/SA] at 8.691E+03 yr |
| I129 | 1.2110E-05 [Ci/yr/SA] at 1.000E+04 yr |
| Cl36 | 5.1927E-06 [Ci/yr/SA] at 1.000E+04 yr |
| Cm246 | 2.7135E-06 [Ci/yr/SA] at 8.101E+03 yr |
| Np237 | 1.7309E-10 [Ci/yr/SA] at 1.000E+04 yr |
| Am241 | 6.0024E-14 [Ci/yr/SA] at 9.543E+03 yr |

59/88

subarea 2 of 7 realization 1 of 1

exec: calling uzflow
exec: calling nfenv
exec: calling ebsfail
*** No Corrosion WP Failure ***
exec: failed WPs from INITIAL event = 17 at time = 0.0 yr
*** failed WPs: 17 out of 1767 ***
exec: calling ebsrel
There is no EBS release
exec: calling uzft
There is no UZ release
exec: calling szft
There is no SZ release

subarea 3 of 7 realization 1 of 1

exec: calling uzflow
exec: calling nfenv
exec: calling ebsfail
*** No Corrosion WP Failure ***
exec: failed WPs from INITIAL event = 8 at time = 0.0 yr
*** failed WPs: 8 out of 855 ***
exec: calling ebsrel
Highest release rates from Sub Area 3
Tc99 9.2848E-03 [Ci/yr/SA] at 9.769E+03 yr
Ni59 1.7539E-03 [Ci/yr/SA] at 9.769E+03 yr
Cl14 3.6029E-04 [Ci/yr/SA] at 9.543E+03 yr
Cs135 2.9074E-04 [Ci/yr/SA] at 9.769E+03 yr
Se79 2.8437E-04 [Ci/yr/SA] at 9.769E+03 yr
Np237 9.8061E-05 [Ci/yr/SA] at 1.000E+04 yr
exec: calling uzft
Highest release rates from UZ
Tc99 9.2733E-03 [Ci/yr/SA] at 9.769E+03 yr
Ni59 1.7517E-03 [Ci/yr/SA] at 9.769E+03 yr
Cs135 2.9040E-04 [Ci/yr/SA] at 9.769E+03 yr
Se79 2.8404E-04 [Ci/yr/SA] at 9.769E+03 yr
Np237 9.8059E-05 [Ci/yr/SA] at 1.000E+04 yr
Cm245 4.3605E-05 [Ci/yr/SA] at 9.769E+03 yr
exec: calling szft
Highest release rates from SZ
Cm245 2.8226E-09 [Ci/yr/SA] at 1.000E+04 yr
I129 8.8129E-10 [Ci/yr/SA] at 1.000E+04 yr
Cl36 6.9318E-10 [Ci/yr/SA] at 1.000E+04 yr
Cm246 2.9274E-10 [Ci/yr/SA] at 1.000E+04 yr
Np237 1.6109E-16 [Ci/yr/SA] at 1.000E+04 yr
Am241 1.0231E-18 [Ci/yr/SA] at 1.000E+04 yr

subarea 4 of 7 realization 1 of 1

exec: calling uzflow
exec: calling nfenv
exec: calling ebsfail
*** No Corrosion WP Failure ***
exec: failed WPs from INITIAL event = 5 at time = 0.0 yr
*** failed WPs: 5 out of 472 ***
exec: calling ebsrel
Highest release rates from Sub Area 4
Tc99 9.1174E-03 [Ci/yr/SA] at 1.723E+03 yr
Ni59 1.7872E-03 [Ci/yr/SA] at 1.723E+03 yr

C14 1.0205E-03 [Ci/yr/SA] at 1.723E+03 yr
 Am241 7.3673E-04 [Ci/yr/SA] at 1.723E+03 yr
 Se79 3.2172E-04 [Ci/yr/SA] at 1.723E+03 yr
 Cs135 3.0260E-04 [Ci/yr/SA] at 1.723E+03 yr

exec: calling uzft

Highest release rates from UZ

Tc99 8.2456E-03 [Ci/yr/SA] at 1.766E+03 yr
 Ni59 1.6161E-03 [Ci/yr/SA] at 1.766E+03 yr
 Am241 6.6946E-04 [Ci/yr/SA] at 1.766E+03 yr
 Se79 2.9094E-04 [Ci/yr/SA] at 1.723E+03 yr
 Cs135 2.7367E-04 [Ci/yr/SA] at 1.723E+03 yr
 Np237 2.1460E-04 [Ci/yr/SA] at 1.766E+03 yr

exec: calling szft

Highest release rates from SZ

Tc99 9.7650E-05 [Ci/yr/SA] at 1.000E+04 yr
 Cm245 8.1937E-06 [Ci/yr/SA] at 3.549E+03 yr
 I129 2.8825E-06 [Ci/yr/SA] at 4.093E+03 yr
 Cm246 1.3297E-06 [Ci/yr/SA] at 3.549E+03 yr
 Cl36 1.3246E-06 [Ci/yr/SA] at 3.549E+03 yr
 Np237 7.8361E-11 [Ci/yr/SA] at 1.000E+04 yr

 subarea 5 of 7 realization 1 of 1

exec: calling uzflow

exec: calling nfenv

exec: calling ebsfail

*** No Corrosion WP Failure ***

exec: failed WPs from INITIAL event = 6 at time = 0.0 yr

*** failed WPs: 6 out of 654 ***

exec: calling ebsrel

There is no EBS release

exec: calling uzft

There is no UZ release

exec: calling szft

There is no SZ release

 subarea 6 of 7 realization 1 of 1

exec: calling uzflow

exec: calling nfenv

exec: calling ebsfail

*** No Corrosion WP Failure ***

exec: failed WPs from INITIAL event = 7 at time = 0.0 yr

*** failed WPs: 7 out of 738 ***

exec: calling ebsrel

There is no EBS release

exec: calling uzft

There is no UZ release

exec: calling szft

There is no SZ release

 subarea 7 of 7 realization 1 of 1

exec: calling uzflow

exec: calling nfenv

exec: calling ebsfail

*** No Corrosion WP Failure ***

exec: failed WPs from INITIAL event = 3 at time = 0.0 yr

*** failed WPs: 3 out of 278 ***

exec: calling ebsrel

Highest release rates from sub Area 7

| | | |
|-------|-----------------------|-----------------|
| Tc99 | 1.0809E-03 [Ci/yr/SA] | at 3.549E+03 yr |
| Ni59 | 2.0964E-04 [Ci/yr/SA] | at 3.549E+03 yr |
| C14 | 9.8848E-05 [Ci/yr/SA] | at 3.549E+03 yr |
| Se79 | 3.7955E-05 [Ci/yr/SA] | at 3.549E+03 yr |
| Cs135 | 3.6383E-05 [Ci/yr/SA] | at 3.549E+03 yr |
| Np237 | 3.2403E-05 [Ci/yr/SA] | at 1.000E+04 yr |

exec: calling uzft

Highest release rates from UZ

| | | |
|-------|-----------------------|-----------------|
| Tc99 | 1.0712E-03 [Ci/yr/SA] | at 3.635E+03 yr |
| Ni59 | 2.0768E-04 [Ci/yr/SA] | at 3.635E+03 yr |
| Se79 | 3.7551E-05 [Ci/yr/SA] | at 3.635E+03 yr |
| Cs135 | 3.6027E-05 [Ci/yr/SA] | at 3.635E+03 yr |
| Np237 | 3.2403E-05 [Ci/yr/SA] | at 1.000E+04 yr |
| Am243 | 1.3633E-05 [Ci/yr/SA] | at 1.000E+04 yr |

exec: calling szft

Highest release rates from SZ

| | | |
|-------|-----------------------|-----------------|
| Tc99 | 5.0896E-05 [Ci/yr/SA] | at 1.000E+04 yr |
| Cm245 | 4.3805E-06 [Ci/yr/SA] | at 6.560E+03 yr |
| I129 | 2.0859E-06 [Ci/yr/SA] | at 8.101E+03 yr |
| Cl36 | 9.2960E-07 [Ci/yr/SA] | at 7.551E+03 yr |
| Cm246 | 5.9295E-07 [Ci/yr/SA] | at 6.113E+03 yr |
| Np237 | 2.6422E-11 [Ci/yr/SA] | at 1.000E+04 yr |

exec: calling dcagw

Highest annual dose GW pathway

| | | |
|-------|----------------------|-----------------|
| Cm245 | 1.7074E-02 [mrem/yr] | at 8.101E+03 yr |
| Cm246 | 2.0989E-03 [mrem/yr] | at 7.551E+03 yr |
| I129 | 1.7803E-03 [mrem/yr] | at 1.000E+04 yr |
| Tc99 | 6.4596E-05 [mrem/yr] | at 1.000E+04 yr |
| Cl36 | 2.7012E-05 [mrem/yr] | at 9.323E+03 yr |
| Np237 | 2.2960E-07 [mrem/yr] | at 1.000E+04 yr |

At end of TPI, annual dose GW pathway

| | |
|-------|----------------------|
| Cm245 | 1.5858E-02 [mrem/yr] |
| I129 | 1.7803E-03 [mrem/yr] |
| Cm246 | 1.7119E-03 [mrem/yr] |
| Tc99 | 6.4596E-05 [mrem/yr] |
| Cl36 | 2.6933E-05 [mrem/yr] |
| Np237 | 2.2960E-07 [mrem/yr] |
| sum | 1.9442E-02 [mrem/yr] |

exec: end realizations

exec: Run Successfully Completed

pcl5

JOB 147

tpa3.2pcport

Summary of
testing &
installation test

R. Rice
11/3/98

For: bigbend!rrice
Date: Tue Nov 3 10:11:12 CST 1998
Submit queue: IF 1 / Ethernet / UHSW
Submitted: Mon Nov 02 12:27:32 1998
Started: Mon Nov 02 12:27:32 1998



QMS 3825 Print System

QMS 3825 Print System

63/88

(summary of tests performed)

Tests performed in the UNIX port code and the TPA 3.2 (dev) Version:

1. basecase tpa.inp
2. basecase tpa.inp for 10 vectors
3. basecase tpa.inp with VOLCANO on
4. basecase tpa.inp with VOLCANO on and 10 vectors
5. basecase tpa.inp with FAULTO on
6. basecase tpa.inp with FAULTO on and 10 vectors
7. basecase tpa.inp with all append files generated
8. basecase tpa.inp with all append files generated and 10 vectors
9. basecase tpa.inp for 250 vectors, 50,000 yr, and all disruptive events on

In the comparison of results, all files matched using the UNIX 'diff' feature.

Installed the TPA 3.2 PC beta code on Cathy's machine, the 486 box in Room A241, and on the Pentium 200 ("cleric") in the lab downstairs using the self-extracting zip files in the 3 diskettes supplied to me by Michael Muller (this version was the same as the one Sitakanta took with him to the NRC and that was installed on the lap top). The files were extracted and the placed into the directory that I selected. I set the path (set TPA_DATA=d:\tpa and set TPA_TEST=d:\tpa) and ran the TPA code. The code ran with the precompiled executables, except on cleric I had to get Jose to increase the virtual memory (swap size) from 100 MB to something much larger (I asked for 400 min and 500 max MB) and then the TPA code successfully ran on that machine and gave output consistent with the other machines and from bigbend (note that the tpa3.2 dev.version of the code needed to be compiled for snllhs.f with -O3 and -r8 to give the same snllhs.f output in lhs.out)

R. Rice
11/3/98

TPA V 3.2 PVM


SOFTWARE RELEASE NOTICE

| | | |
|--|----------------------|--------------------------|
| 1. SRN Number: PA-SRA-211 | | |
| 2. Project Title: TSPA & Technical Integration Code | | Project No. 20-01402-762 |
| 3. SRN Title: TPA Version 3.2 PVM β Rf 10-4-99 | | |
| 4. Originator/Requestor: Ron Janetzke | | Date: Oct. 1, 1999 |
| 5. Summary of Actions <ul style="list-style-type: none"> <input type="checkbox"/> Release of new software <input type="checkbox"/> Release of modified software: <ul style="list-style-type: none"> <input type="checkbox"/> Enhancements made <input type="checkbox"/> Corrections made <input type="checkbox"/> Change of access software <input checked="" type="checkbox"/> Software Retirement | | |
| 6. Persons Authorized Access | | |
| Name | Read Only/Read-Write | Addition/Change/Delete |
| Ron Janetzke | RW | |
| Sitakanta Mohanty | RW | |
| Tim McCartin (NRC) | RW | |
| M. Rose Byrne (NRC) | RW | |
| 7. Element Manager Approval: <i>Gordon Williams</i> | | Date: 10/4/99 |
| 8. Remarks: Version superceded. | | |

SOFTWARE SUMMARY FORM


| | | | | | |
|---|--|---|--|--|--|
| 01. Summary Date: 11/23/98 | | 02. Summary prepared by (Name and phone): Ron Janetzke (210) 522-3318 | | 03. Summary Action: Modified | |
| 04. Software Date: 11/23/98 | | 05. Short Title: TPA Version 3.2 PVM beta | | | |
| 06. Software Title: Total-System Performance Assessment 3.2 with Parallel Virtual Machine (beta) | | | | 07. Internal Software ID: None | |
| 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 type="checkbox"/> Scientific/Engineering <input type="checkbox"/> Auxiliary Analyses <input checked="" type="checkbox"/> Total System PA <input type="checkbox"/> Subsystem PA <input type="checkbox"/> Other b. Specific: | |
| 11. Submitting Organization and Address: CNWRA/SwRI 6220 Culebra Road San Antonio, TX 78228 | | | | 12. Technical Contact(s) and Phone: Sitakanta Mohanty (210) 522-5185 | |
| 13. Software Application: The TPA Code consists of the following modules: UZFLOW, NFENV, EBSREL, UZFT, SZFT, DCAGW, FAULTO, SEISMO, VOLCANO, ASHPLUMO, ASHRMVO, DCAGS, LHS, EXEC. The PARJOB Code includes: get_result.f, parjob_loc.f, run_job.f, tpa_loc.f, and write_result.f, among others. | | | | | |
| 14. Computer Platform: SUN Workstation | | 15. Computer Operating System: UNIX | | 16. Programming Language(s): FORTRAN | |
| 17. Number of Source Program Statements: Approx. 32,000 lines w/o Stand Alone Codes | | 18. Computer Memory Requirements: 70 Mb | | 19. Tape Drives: None | |
| 20. Disk Units: N/A | | 21. Graphics: N/A | | 22. Other Operational Requirements: PVM environment required. | |
| 23. Software Availability: <input type="checkbox"/> Available <input checked="" type="checkbox"/> Limited <input type="checkbox"/> In-House ONLY | | | 24. Documentation Availability: <input type="checkbox"/> Available <input type="checkbox"/> Preliminary <input checked="" type="checkbox"/> In-House ONLY | | |
| 25. Software Developer: <i>Ron Janetzke</i> Date: <i>11-24-98</i> | | | | | |

SOFTWARE RELEASE NOTICE

| | | |
|---|----------------------|-------------------------|
| 1. SRN Number: PA-SRN-179 | | |
| 2. Project Title: PVM Version of TPA 3.2 | | Project No. 20-1402-762 |
| 3. SRN Title: TPA 3.2 PVM beta | | |
| 4. Originator/Requestor: Bruce Mabrito | | Date: 11/23/98 |
| 5. Summary of Actions <div style="margin-left: 20px;"> <input type="checkbox"/> Release of new software <input checked="" type="checkbox"/> Release of modified software: <div style="margin-left: 20px;"> <input checked="" type="checkbox"/> Enhancements made <input type="checkbox"/> Corrections made <input type="checkbox"/> Change of access software <input type="checkbox"/> Software Retirement </div> </div> | | |
| 6. Persons Authorized Access | | |
| Name | Read Only/Read-Write | Addition/Change/Delete |
| Sitakanta Mohanty | RW (TPA only) | Addition |
| Ron Janetzke | RW (TPA only) | Addition |
| Tim McCartin (NRC) | RW (TPA only) | Addition |
| M. Rose Byrne (NRC) | RW (TPA only) | Addition |
| 7. Element Manager Approval:  | | 11/24/98 Date: |
| 8. Remarks: | | |

SOFTWARE CHANGE REPORT (SCR)

An SCR must be filled up for reporting any problems found in a developed code or any necessary modifications that must be brought into the code. A revised Software Requirement Document (SRD) is needed if extensive modifications to the code are expected. If one SCR description of a group of small changes, it will be considered acceptable if specific changes are clearly identified with the modified modules.

| | | |
|---|--|-----------------------------------|
| SCR No. (Software Developer Assigns): PA-SCR-258 | Software Title and Version: TPA3.2 PVMbeta | Project No: 20-1402-762 |
| Affected Software Module(s), Description of Problem(s): tpa-loc.f run-job.f Modules reference a hardcoded location for tpa.e, and will not run on a host other than big bend. | | |
| Change Requested by: millwater Date: 12-1-98 | Change Authorized by (Software Developer):  Date: 12/3/98 | |
| Description of Implemented Change(s) (If changes not implemented, please justify): Routines were changed to always use the tpa.e path specified in tpa-loc.f. | | |
| Implemented by: N. Kelly | Completed Date: 12-3-93 | |

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE: TPA Version 3.2 Parallel Virtual Machine (PVM) Beta

November 23, 1998

Total-System Performance Assessment (Scientific and Engineering Software) Version 3.2 Parallel Virtual Machine (PVM) Beta

NOTE: This version of the TPA Software contains the integration of TPA Version 3.2 with PARJOB which controls the execution of TPA and prepares it to execute on several machines simultaneously. An electronic scientific notebook has been utilized as the change documentation method and is being retained in the TPA Version 3.2 PVM Beta folder.

1. **This Design Verification Report is prepared by:** Bruce Mabrito in conjunction with Ron Janetzke.
Full Title of CNWRA scientific and engineering software: Total-System Performance Assessment (TPA) Version 3.2 Parallel Virtual Machine (PVM) Beta.
Demonstration work station: SPARC20 in conjunction with the SCRATCHY1 server from R. Janetzke's office.
Operating System: SunOS 5.5.1

2. **Software Requirements Description and any changes thereto approved by Element Manager?**

YES
NO
N/A

NOTE: A very straightforward and short SRD was prepared by H. Millwater (of SwRI Division 06) and was signed after-the-fact by the CNWRA PA Element Manager.

3. **Software Development Plan (SDP) and any changes have been approved by the Element Manager?**

YES
NO
N/A

NOTE: A very straightforward and short SDP was prepared by H. Millwater and was signed after-the-fact by the CNWRA PA Element Manager.

4. **Design and Development**
Module-level testing is documented in either scientific notebooks or in Software Change Reports?

YES
NO
N/A

NOTE: Note: An electronic scientific notebook (uncontrolled) was utilized and contains module level documentation. A copy of the printed notebook is in the TPA Version 3.2 PVM Beta file.

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

YES

NO

N/A

NOTE: PARJOB was acquired from SwRI to facilitate merging/integrating TPA V 3.2 PVM Beta into a useable code. PARJOB is proprietary SwRI software and the CNWRA does not and will not have a copy of the source code..

6. Is the CNWRA software documented internally?

YES

NO

N/A

NOTE: See PARJOB routines. Modification of TPA V 3.2 software did occur so that it can be integrated with the PARJOB software.

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: The latest CNWRA/SwRI Contract No. (NRC-02-97-009) was not reflected in the source code module header of TPA Version 3.2 PVM Beta. The software developer was made aware of this but it was agreed not to change it at this late date in the development of the code. Other requirements were fulfilled.

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: The TPA SwRI Div. 06 Version 3.2 PVM Beta Users' Manual (PARJOB, dated Nov. 20, 1998) was available during the Design Verification activities. A separate TPA V 3.2 PVM Beta handout will be written by the CNWRA and sent to the NRC later.

Are there basic instructions for the use of the software?

YES

NO

N/A

NOTE: The CNWRA will produce a TPA V 3.2 PVM Beta handout for issuance with the software.

10. Acceptance Testing

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

☒ YES☐ NO☐ N/A

NOTE: TPA V3.2 PVM Beta was compiled, linked, executed and tested on BigBend. Since the CNWRA has no direct access to CRADAL (the future NRC server), that testing could not be conducted from the San Antonio location.

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

☒ YES☐ NO☐ N/A

NOTE: Acceptance testing on Sun platforms with the Solaris O.S. was performed. Summaries are in the electronic scientific notebook pages in this folder.

Have installation tests been performed on the target platform?

☒ YES☐ NO☐ N/A

NOTE: Tests have been performed on the BIGBEND server.

11. Configuration Control

Is the Software Summary Form completed and signed?

☒ YES☐ NO☐ N/A

If no, explain:

12. Is a software technical description prepared, documenting the essential mathematical and numerical basis?☒ YES☐ NO☐ N/A

If no, explain: The technical description is given in the Users' Manual for PARJOB, the SwRI Parallel Processing Program.

13. Is the source code available (or, is the executable code available in the case of commercial codes)?☐ YES☒ NO☐ N/A

NOTE: For the TPA V 3.2 PVM Beta the answer is yes. For PARJOB, the answer is no.

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

NOTE: The source code for TPA V 3.2 PVM Beta and the PARJOB make files have been submitted to CNWRA QA.

 11/23/98

Ron Janetzke Date

CNWRA TPA Software Co-Developer

Attachments/

Original to: Software Folder

cc: CNWRA Software Developer
Cognizant EM
S. Mohanty

 11/23/98

Bruce Mabrito Date

CNWRA Software Custodian

REQUIREMENT DESCRIPTION

SOFTWARE REQUIREMENTS DESCRIPTION - TPA Version 3.2PVMbeta (Parallel Machines)

1.0 SOFTWARE FUNCTION

The PVM (Parallel Virtual Machine) library will be used to network UNIX workstations into a pseudo parallel processing machine that performs TPA analysis in parallel, thus reducing computational time.

2.0 TECHNICAL BASIS: PHYSICAL AND MATHEMATICAL MODEL

The features of PVM will be integrated with the TPA Version 3.2 code in order to compute the results for a series of input vectors in a more computationally efficient manner by performing the computations in parallel. The PVM approach utilizes existing hardware, UNIX workstations, and existing software. Additionally, much of the PVM coordinating FORTRAN code has already been developed at SwRI under Internal Research funding.

3.0 COMPUTATIONAL APPROACH

3.1 Data Flow and User Interface

(not applicable - the same as the TPA Version 3.2 code)

3.2 Hardware and Software Requirements

- **Target platform**
SUN (the same as TPA Version 3.2 code)
- **Operating system**
Solaris (the same as TPA Version 3.2 code)
- **Programming language**
FORTRAN 77 (the same as TPA Version 3.2 code)

3.3 Graphics Requirements

(not applicable)

3.4 Pre- and Post-Processors

(not applicable)

4.0 REFERENCES

(not applicable)

APPENDICES - (not applicable)

Approval: *Gerhard Wittmayer* 11/23/98

SOFTWARE DEVELOPMENT PLAN

Software Development Plan - PARJOB, Parallel Processing Program for TPA Version 3

1.0 Scope

The PARJOB parallel processing software will be customized to interface with the Total Performance Assessment (TPA) version 3.2 software. PARJOB presently interfaces with TPA version 2.0.

The original "core" software subroutines have been developed on SwRI internal research funding, e.g., 06-9994. These routines will be delivered in a compiled form but source code will not be delivered. Routines specific to interfacing with TPA version 3.2 will be delivered in source form.

2.0 Baseline items

One software product will be produced - a customized version of PARJOB which works with TPA 3.2. One user's manual will be produced. A number of example problems will be produced

3.0 Project Management

3.1 Work Breakdown Structure (man-weeks in parentheses)

- Task 1 - install PARJOB on Sun Solaris system and verify. (1-2)
- Task 2 - modify TPA executive.(2-3)
- Task 3 - modify slave to call TPA for a single vector analysis.(2-3)
- Task 4 - modify slave to read all .res files and return results to Master.(4-6)
- Task 5 - modify master to concatenate all results into cumulative .res files.(4-6)
- Task 6 - verification.(2-4)
- Task 7 - documentation.(1-2)

3.2 Projected Schedule (start-end date)

- Task 1 - 7/6/98-7/13/98
- Task 2 - 8/3/98-8/8/98
- Task 3 - 8/8/98-8/15/98
- Task 4 - 8/15/98-9/1/98
- Task 5 - 9/1/98-9/21/98
- Task 6 - 9/21/98-9/31/98
- Task 7 - 9/21/98-9/31/98

3.3 Staffing

The work will be carried out by Harry Millwater - Principle Engineer, Div 06, and coop students Clay Cowan and Nathan Kelley both of Div 06.

3.4 Risk Management

The significant element of risk lies in the reading of the .res files on the Slaves, the return of their data, and the reconstruction of the completed .res files on the Master. PARJOB has been designed to return only one number from the Master to the Slave. PARJOB will now have to read several lines of results from several files. We feel this can be accomplished by looping over the files, reading the results with character strings from each file, passing back the strings to the Master, storing the strings until all runs are completed, then reconstructing the total .res files.

A character string approach will hopefully handle the difficulties with file formats. We do not want to have to know the formats of each file. We want to read the results as a character string, return the string to the Master, and reconstruct all the strings into a completed results file.

4.0 Development Procedures

4.1 Hardware and Software Resources

The deployment platform is Sun Solaris Unix workstations. Access to a networked Sun system is necessary.

All programming is done using Fortran 77 and the Parallel Virtual Machine free public domain software. Fortran is used for all logic and coordination tasks, and PVM is used for all message passing between the Master and the Slaves. The only requirements are for a Fortran 77 compiler.

4.2 Software Development Lifecycle

Because of the relatively small development effort involved in this project, and the considerable software brought to the project, prototype, alpha, beta, etc. stages of the software were not necessary.

Task 7 - documentation.(1-2)

4.3 Coding

All programming is done using Fortran 77. The Parallel Virtual Machine free public domain software is used for all message passing between the Master and the Slaves.

4.4 Acceptance Testing and Analysis

Acceptance testing is straightforward in this case. The parallel processing analysis must result in the identical results files as the standard sequential analysis.

A series of test problems will be developed and run. A base case will be identified and a number of perturbations of inputs from the base case will be run.

The public domain Fortran checker "ftnchek" will be used to perform a static analysis on the software to identify any Fortran semantic errors.

5.0 Configuration Management Plan (CMP)

5.1 Tools

The UNIX Source Code Control System (SCCS) will be used for configuration control on *scratchyl* running SOLARIS with SPARCworks.

5.2 Configuration Identification

Not all of the TPA source code will be maintained in the PVM SCCS system. Rather only routines that were changed to implement the PVM version will be controlled by PVM SCCS. Other routines will be controlled by the regular TPA SCCS procedures.

For PARJOB only code written specifically for the NRC implementation of the PVM version of PARJOB will be performed on *scratchyl*. All other PARJOB routines will be controlled

in Division 06. The SCCS version number is assigned on an automatic basis and may not necessarily agree with the version number specified under the NRC contract. Control will begin with SCCS version 1.1 upon successful completion of the first acceptance test.

5.3 Configuration Procedures

All check-in or check-out activities on *scratchy1* will be performed by Ron Janetzke. The latest version of the files will be available in /export/home/janetzke/tpa/pvm.

The standard SCR change request form will be used for all changes to the controlled source code.

Backups will be performed by the system administrator of the host machines.

6.0 References

Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Mancheck, R., Sunderam, V., "PVM: Parallel Virtual Machine, A User's Guide and Tutorial for Networked Parallel Computing," MIT Press, Cambridge, Massachusetts, London, England, 1994

7.0 Appendices

Approved: *Arden Wittmeyer* 11/23/98

c Program Name: TPA - Total-System Performance Assessment Code RJOB

PARallel processing of JOBS
Version 1.0
Southwest Research Institute
September 1998

DATE: 11-23-1998 15:14

DATE/Time "Stamp"
Version #

Is this a restart? (y)es or (n)o?

temp

[illegible]

DISCLAIMER

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 the program (including but not limited to loss of
c data or data being rendered inaccurate or losses sustained by third
c parties or a failure of the program to operate with other programs),
c even if you have been advised of the possibility of such damages or
c for any claim by any other party."

c by S. Mohanty, R. Janetzke, R. Rice, A. Lozano
c R. Manteufel (initial version)

temp

SCIENTIFIC NOTEBOOK INFORMATION

16/63

H. Millwater

SCIENTIFIC NOTEBOOK

Printed: November 13, 1998

INITIALS: HM

SCIENTIFIC NOTEBOOK

by

Harry Millwater

Harry Millwater

Southwest Research Institute
San Antonio, Texas

H. Millwater

SCIENTIFIC NOTEBOOK

Printed: November 13, 1998

INITIALS: HM**INITIAL ENTRIES**

Scientific Notebook:

Issued to: S. Mohanty

Issue Date: Nov. 13, 1998

Account Number: 20-1402-760

Title: PARJOB, Parallel Processing Program for TPA Version 3

Participants: Harry Millwater, Clay Cowan, Nathan Kelley

July 13, 1998 - Initial Port of PARJOB code from HP to Sun

In preparation of customizing PARJOB to work with TPA version 3, Clay ported the existing PARJOB code from the HP to the Sun. This version worked correctly with TPA version 2. He also installed the latest version of PVM (3.4beta6) on the Sun. The port went relatively smoothly once Clay got acclimated to the Sun operating system. The code was verified using a simple function evaluation not TPA.

July 29, 1998 - Meeting with Ron Janetzke to outline modification needed for TPA version 3.2. Ron described the TPA version 3 code and we discussed a potential implementation approach whereby internal variables in TPA would be packed and sent to the slaves and unpacked inside TPA on the slaves. This approach appears difficult to implement because many arrays need to be packed and unpacked, which is prone to error.

July 31, 1998 - Discussed complexity of current approach with Sitikanta. He agreed to let he and Ron discuss and see if they can come up with a better approach.

August 3, 1998 - Sitikanta and Ron agreed to modify TPA version 3.2 so that it could be run analogous to TPA version 2. The idea of passing many internal variables is dropped in favor of using files. Ron will restructure TPA so that it can be run in 3 modes: mode 1 to generate the lhs samples on Master, mode 2 to run read the samples from a file and analyze on Slave, mode 3 to generate the ccdf-related results files on the Master. This is a very nice approach in that we (Div 06) do not need to modify TPA and the interfaces between PARJOB and TPA are very clean.

The major effort on PARJOB is to modify it to return multiple results files from the Slaves to the Master instead of a single number as done with TPA 3.2. This requires significant changes to subroutines run_job to loop over and read the results files, get_result to store file results in direct access files, and write_results to reconstruct *.res files after all runs are complete. A strategy using character variables to read the results is proposed. That way, no understanding of the

results file formats is required.

August 10, 1998 - Character variable approach to reading results files appears to work. Major effort is to account for a variable number of lines to be returned from the Slave to the Master.

August 20, 1998 - Clay returning to school. Majority of the code working. Most of the results files get returned correctly to the Master and reconstructed.

August 28, 1998 - Nathan Kelley here to replace Clay. Begins work on debugging code.

September 21, 1998 - Writing User's Manual for PARJOB-TPA3.2. Nathan implements 20 verification problems after discussing with Ron Janetzke.

September 29, 1998 - All verification runs give identical results compared with sequential runs.

November 2, 1998 - Repackaging of PARJOB for distribution to NRC. Cleaned up User's Manual. Set up simplified make file.

November 12, 1998 - Set off verification problems again using TPA distribution.

November 13, 1998 - All verification problems run correctly.

Static Analysis run using ftnchek All routines pass without errors. Subroutines STCHAR and STCHR2 are general purpose parsing routines which accept a character string into a character array. Thus, the warnings regarding argument type mismatches.

FTNCHEK Version 2.8 May 1995

```
File check_dir.f:
Including file fpvm3.h:
  0 syntax errors detected in file check_dir.f
File cleanup.f:
Including file fpvm3.h:
  0 syntax errors detected in file cleanup.f
File dater.f:
  0 syntax errors detected in file dater.f
File get_det_data.f:
Including file fpvm3.h:
  0 syntax errors detected in file get_det_data.f
File get_file_name.f:
  0 syntax errors detected in file get_file_name.f
File get_jobs_data.f:
Including file fpvm3.h:
  0 syntax errors detected in file get_jobs_data.f
File get_result.f:
Including file fpvm3.h:
Including file file_names.i:
  0 syntax errors detected in file get_result.f
File job_split.f:
  0 syntax errors detected in file job_split.f
File mod_queues.f:
  0 syntax errors detected in file mod_queues.f
File parjob.f:
```

```
Including file fpvm3.h:
Including file file_names.i:
Warning in module PARJOB file parjob.f:
  Variables set but never used:
    NAME          SKIP
```

```
0 syntax errors detected in file parjob.f
1 warning issued in file parjob.f
```

```
File parjob_loc.f:
  0 syntax errors detected in file parjob_loc.f
File pvmprep.f:
  0 syntax errors detected in file pvmprep.f
File read_input.f:
  0 syntax errors detected in file read_input.f
File read_send_det_data.f:
  Including file fpvm3.h:
    0 syntax errors detected in file read_send_det_data.f
File restart.f:
  0 syntax errors detected in file restart.f
File return_dead.f:
  Including file fpvm3.h:
    0 syntax errors detected in file return_dead.f
File run_job.f:
  Including file fpvm3.h:
  Including file file_names.i:
    0 syntax errors detected in file run_job.f
File send_smp_data.f:
  Including file fpvm3.h:
    0 syntax errors detected in file send_smp_data.f
File spawn_procs.f:
  Including file fpvm3.h:
    0 syntax errors detected in file spawn_procs.f
File stchar.f:
  0 syntax errors detected in file stchar.f
File stchr2.f:
  0 syntax errors detected in file stchr2.f
File summarize_jobs.f:
  0 syntax errors detected in file summarize_jobs.f
File tpa_loc.f:
  0 syntax errors detected in file tpa_loc.f
File write_result.f:
  Including file file_names.i:
    0 syntax errors detected in file write_result.f
```

```
Subprogram STCHAR:  argument arrayness mismatch
  at position 1:
    Dummy arg STRING is whole array in module STCHAR line 1 file stchar.f
    Actual arg NAME(I) is array element in module WRITE_RESULT line 43 file
write_result.f
Subprogram STCHAR:  argument arrayness mismatch
  at position 1:
    Dummy arg STRING is array in module STCHAR line 1 file stchar.f
    Actual arg STORE is scalar in module RUN_JOB line 184 file run_job.f
Subprogram STCHAR:  argument arrayness mismatch
  at position 1:
    Dummy arg STRING is array in module STCHAR line 1 file stchar.f
    Actual arg PROPID is scalar in module RESTART line 66 file restart.f
Subprogram STCHAR:  argument arrayness mismatch
  at position 1:
    Dummy arg STRING is whole array in module STCHAR line 1 file stchar.f
    Actual arg NAME(L) is array element in module GET_RESULT line 298 file
get_result.f
```

20/63

HM

```
Subprogram STCHAR:  argument arrayness mismatch
  at position 1:
    Dummy arg STRING is array in module STCHAR line 1 file stchar.f
    Actual arg PROPID is scalar in module GET_FILE_NAME line 79 file
get_file_name.f
Subprogram STCHR2:  argument arrayness mismatch
  at position 1:
    Dummy arg STRING is whole array in module STCHR2 line 1 file stchr2.f
    Actual arg NAME(I) is array element in module WRITE_RESULT line 44 file
write_result.f
Subprogram STCHR2:  argument arrayness mismatch
  at position 1:
    Dummy arg STRING is array in module STCHR2 line 1 file stchr2.f
    Actual arg STORE is scalar in module RUN_JOB line 185 file run_job.f
```


*To: TPA 3.2 PVM
Software Folder*

**INSTALLATION AND EXECUTION OF PARALLEL VIRTUAL
MACHINE (PVM) VERSION OF THE TOTAL-SYSTEM
PERFORMANCE ASSESSMENT (TPA) VERSION 3.2 CODE**

Prepared for

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

Prepared by

**Harry Millwater
Nathan Kelley
Clay Cowan
Ron Janetzke
Sitakanta Mohanty**

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

December 1998

ACKNOWLEDGMENTS

This work was performed on behalf of the NRC Office of Nuclear Material Safety and Safeguards, Division of Waste Management under contract No. NRC-02-97-009. This document is an independent product of the CNWRA and does not necessarily reflect the view or regulatory position of the NRC.

The TPA3.2PVMbeta code has been developed following the procedures described in the CNWRA Technical Operating Procedure, TOP-018, which implements the QA guidance contained in the CNWRA QA Manual.

The authors thank Gordon Wittmeyer and Wes Patrick for their reviews of the report. The authors are thankful to Cathy Garcia and Annette Mandujano for their typing and formatting help in preparing the document.

CONTENTS

| Section | Page |
|--|------|
| ACKNOWLEDGMENTS | iii |
| FIGURES | vii |
| TABLES | ix |
| 1 INTRODUCTION | 1 |
| 2 FEATURES OF GENERALIZED PARJOB CODE | 1 |
| 3 APPROACH TO PVM IMPLEMENTATION IN TPA 3.2 CODE | 2 |
| 3.1 STEP 1: GENERATE LHS SAMPLE | 2 |
| 3.2 STEP 2: COMPUTE AND STORE RESULTS FOR EACH VECTOR | 4 |
| 3.3 STEP 3: CREATE *.res RESULTS FILES | 5 |
| 4 INSTALLATION OF PARJOB CODE AND CUSTOMIZATION | 6 |
| 4.1 INSTALLATION OF CURRENT VERSION OF PVM | 6 |
| 4.2 UPGRADING TO FUTURE VERSIONS OF PVM | 9 |
| 5 RUNNING PARJOB WITH TPA | 9 |
| 5.1 SETTING UP THE PARALLEL VIRTUAL MACHINE AND WORKING DIRECTORIES | 9 |
| 5.2 INVOKING PARJOB | 11 |
| 5.3 SETTING THE NUMBER OF SLAVE PROCESSES | 11 |
| 5.4 HALTING PVM | 11 |
| 5.5 SUMMARY FILE | 11 |
| 6 RESTRICTIONS | 11 |
| 6.1 HOMOGENEOUS SYSTEMS | 11 |
| 6.2 .rhosts FILE | 12 |
| 6.3 HARD DISK ACCESS | 12 |
| 6.4 UNIQUE WORKING DIRECTORY | 12 |
| 6.5 NUMBER OF SLAVE PROCESSES | 12 |
| 6.6 ANALYZE ALL VECTORS | 12 |
| 7 TROUBLESHOOTING | 12 |
| 7.1 READING SLAVE RESULTS | 12 |
| 7.2 ERROR RUNNING PVM | 13 |
| 7.3 CAN'T ADD HOST | 13 |
| 7.4 TPA ABORTS ON SLAVE | 13 |
| 8 VERIFICATION | 13 |
| 9 REFERENCES | 13 |

FIGURES

| Figure | Page |
|--------|--|
| 1 | Flow chart showing interaction between PARJOB and TPA in a master-slave paradigm 3 |

TABLES

| Table | | Page |
|-------|--|------|
| 1 | *.res files returned to master | 5 |
| 2 | Files in PARJOB distribution set | 6 |
| 3 | Test case parameter values | 14 |

1 INTRODUCTION

The Total-system Performance Assessment (TPA) Version 3.2 code was recently developed by CNWRA and NRC staff for conducting performance assessments of the proposed high-level radioactive waste repository at Yucca Mountain (YM). This probabilistic code performs process-level calculations hundreds or thousands of times using a stratified random sampling method to account for uncertainties in parameters and variables. Ordinarily, the calculations are done serially, implying that the process level computation must be conducted sequentially, realization by realization, using a single processor in a computer. Because a large number of realizations are often required for analysis purposes, the computation time could be prohibitive. One of the alternatives is to be able to run the code on several compatible platforms simultaneously using Parallel Virtual Machine (PVM) methodology (Geist et al., 1994) that has the potential to significantly reduce the computation time. PVM is a free public domain software package from Oakridge National Labs and is used for message passing among the hosts comprising the virtual machine. PVM is available for almost all Unix operating systems.

Southwest Research Institute, through a focused internal research and development project, had developed a generalized software known as PARJOB (PARallel processing of JOBs) (Millwater et al., 1998) written in Fortran 77 and using the PVM library that provides a convenient method for running a computer code that requires repeated sampling. PARJOB is a portable code and has been tested successfully on HP, Sun, and Silicon Graphics workstations.

While the PVM method can in general run a code in a parallel configuration, the benefit of parallel processing strictly depends on the problem to be solved. The TPA code uses the Latin Hypercube Sampling (LHS) method to sample parameters for conducting Monte Carlo simulations. Because several parameters in the code are correlated with each other, the TPA code samples all parameters *a priori* for all realizations prior to initiating any process-level calculations. *A priori* sampling in the TPA code and independence of the process-level calculations from other realizations makes the TPA code amenable to being executed using PVM method.

This document first provides a brief overview of the PARJOB code, followed by the description of several modifications made to the TPA version 3.2 code for running it in parallel mode. Finally, instructions on running the PVM version of the TPA code are provided.

2 FEATURES OF GENERALIZED PARJOB CODE

PARJOB uses a master-slave paradigm for probabilistic analysis. This technique uses a master process on one machine to create, control, and monitor slave processes running on remote machines. Although many slave processes may be created (spawned) by the master, each slave only recognizes one other process in the system, that is the master process. This approach is ideal for LHS because the same model is executed with different inputs. Each analysis of a vector is performed in a slave process and does not need to communicate with other slaves. The vector inputs are sent from the master to the slave and the response is sent from the slave back to the master. PARJOB has a number of fault-tolerant features that are summarized as follows.

- Detect when a slave or master process or host abnormally terminates
- Detect an incorrect working directory
- Detect when a job aborts
- Place a time limit on analysis
- Rerun a job that has failed once, but not twice
- Record results as soon as completed
- Restart from a partially completed analysis

3 APPROACH TO PVM IMPLEMENTATION IN TPA 3.2 CODE

The flow chart showing the interaction between PARJOB and TPA gives an overview of the procedure (figure 1). The TPA code was modified to execute in three different modes such that the analysis can be performed in three sequential steps:

- Step 1: generate LHS sample
- Step 2: compute and store results for each vector
- Step 3: create *.res result files

Steps 1 and 3 are performed by running TPA controlled by the master process. Step 2 is performed one vector at a time by the slave processes, with the results being sent to the master. Modes refer to the execution of the TPA code using only a subset of its functionality. The three modes are as follows:

- Execute up to the point where the *lhs.out* file is generated
- Execute the process level routines for all subareas of one vector
- Execute the routines that generate the cdf files

3.1 STEP 1: GENERATE LHS SAMPLE

The PARJOB internal subroutine *pvmprp* modifies the standard *tpa.inp* input file to instruct TPA to execute step 1. TPA is run in mode 1 to generate the LHS vectors in a file called *lhs.out*. TPA then exits immediately and returns control to PARJOB. During the execution of this step, the TPA code only performs LHS sampling and does not perform any process-level calculations. It should be noted that the Monte Carlo simulation mode in the TPA code should not be selected for PVM runs since this sampling technique cannot simply be distributed over several machines. PARJOB adds the following parameter to *tpa.inp* and sets the parameter to 1 to execute this step:

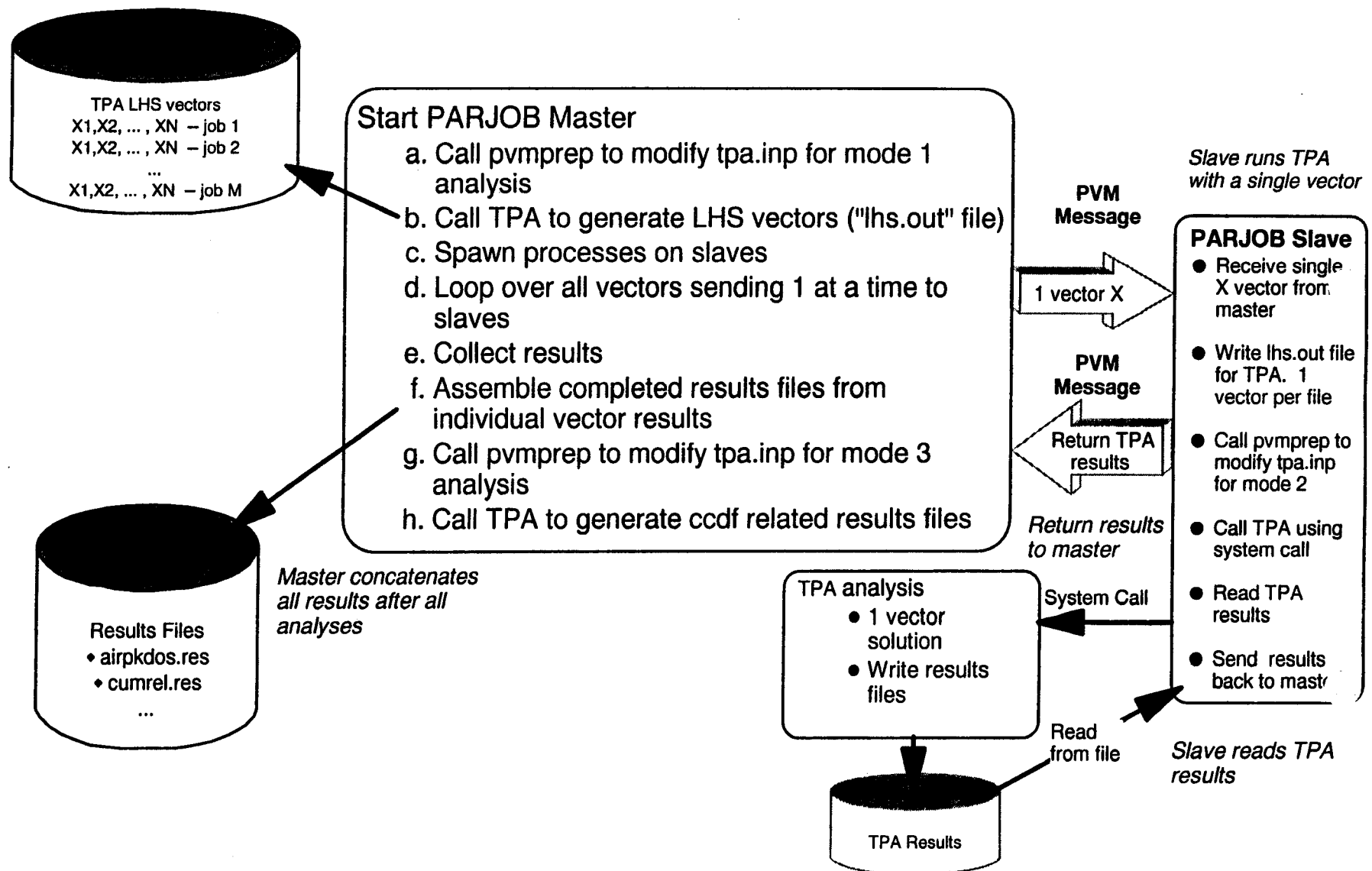


Figure 1. Flow chart showing interaction between PARJOB and TPA in a master-slave paradigm


```
IFLAG
PVMPPREPROCESSOR(Yes=1, No = 0)
1
```

While executing step 1, the start and stop realizations in *tpa.inp* are set to 1 and the total number of vectors to be simulated is set to the desired value.

3.2 STEP 2: COMPUTE AND STORE RESULTS FOR EACH VECTOR

Within the slave process, *pvmpprep* is called to modify the *tpa.inp* file to instruct the TPA code to use mode 2 to execute step 2 and compute only for 1 vector. The input parameters and associated PVM flags for the particular vector to be processed are passed to the slave by the master. The slave writes a local *lhs.out* file which contains only the vector that is passed from the master to the slave. The TPA code runs the single vector input and generates all *.res* output files except for ccdf-related *.res*, which require results from all vectors.

The parameters in *tpa.inp* representing the number of realizations, beginning and ending realizations are set to the vector number:

```
NumberOfRealization = vector #
StartAtRealization = vector #
StopAtRealization = vector #
```

where vector # is the number of the vector to be analyzed. In addition, the following new parameter is introduced into the *tpa.inp* file:

```
**
IFLAG
PVM (Yes=1, No = 0)
1
```

This parameter indicates to the TPA code that it is to perform just the process level calculations in the TPA system. The LHS calculations are not invoked and the CCDF calculations are not performed.

Results from the single-vector TPA analyses are returned to the master for storage and ultimately assembled into the final results files in step 3. PARJOB has been designed so that it does not need to know the format of the output files. This is very significant considering that there are 20 output files with different formats that must be read and returned. In addition, no changes to PARJOB are required should the format of the results files be modified. The **.res* filenames returned to the master are contained in file *filenames.i*.

On the slave, PARJOB reads each line in the results file into a character variable. The number of lines and the character variables are sent back to the master for each *.res* file contained in *filenames.i*. The master unpacks the data and stores the results into a direct access file. The direct access files are used internally and each is titled according to its corresponding file using the *.da* extension (e.g., *cumrel.da*). The output files listed in table 1, which contain information for one vector, are returned to the master. These output files are described in detail in the TPA 3.2 User's Guide (Mohanty and McCartin, 1998).

Table 1. *.res files returned to master

| | |
|-----------------------|-----------------------|
| <i>airpkdos.res</i> | <i>npkdoset.res</i> |
| <i>airpkdos_c.res</i> | <i>npkdoset_c.res</i> |
| <i>ashout.res</i> | <i>pkreltim.res</i> |
| <i>cumrel.res</i> | <i>pkreltim_c.res</i> |
| <i>cumrel_c.res</i> | <i>relgwgs.res</i> |
| <i>gwpkdos.res</i> | <i>relgwgs_c.res</i> |
| <i>gwpkdos_c.res</i> | <i>samplpar.res</i> |
| <i>gwtuzsz.res</i> | <i>totdose.res</i> |
| <i>infilper.res</i> | <i>totdose_c.res</i> |
| <i>nearfld.res</i> | <i>wpsfail.res</i> |

3.3 STEP 3: CREATE *.res RESULTS FILES

Once all vectors have been analyzed, the master process reads the direct access file corresponding to each result file (e.g., *cumrel.da*) and reconstructs the cumulative results file for all vectors (e.g., *cumrel.res*).

Once the non-CCDF related results files are created, TPA is run in mode 3 with the PVMPOSTPROCESSOR flag set to 1 in order to create the CCDF related results file (e.g., *gsccdf.res*). Subroutine *pvmpprep* is called first to modify the *tpa.inp* input file, then TPA is called. TPA will read the existing *.res files and generate the CCDF related results files.

The CCDF-related files generated by TPA in step 3 are:

gsccdf.res
gsccdf_c.res
gwccdf.res
gwccdf_c.res
relccdf.res
relccdf_c.res

The changes made by *pvmpprep* to *tpa.inp* prior to running TPA are:

```
NumberOfRealization = 0
StartAtRealization = 1
StopAtRealization = 1
```

In addition, the following is inserted at the end of *tpa.inp*:

```
**
IFLAG
PVMPOSTPROCESSOR(Yes=1, No=0)
1
```

The PVMPOSTPROCESSOR flag instructs the TPA code to skip the LHS and process level calculations, and to reconstruct the internal arrays necessary for the generation of the CCDF *.res files. The data for these arrays are read from the *.res files reconstructed by PARJOB.

4 INSTALLATION OF PARJOB CODE AND CUSTOMIZATION

4.1 INSTALLATION OF CURRENT VERSION OF PVM

PARJOB can be installed into any convenient directory. A list of the files in the PARJOB distribution is given in table 2. The files included with the distribution set are the source code of routines that are specific to the TPA implementation of PARJOB, associated "include" files, and a link library for the main routines. The user must modify *parjob_loc.f* to indicate the location of the PARJOB executable and *tpa_loc.f* to indicate the location of the TPA executable as described below.

Table 2. Files in PARJOB distribution set

| File | Description |
|-----------------------|---|
| <i>parjob.a</i> | library containing the core PARJOB routines |
| <i>filenames.i</i> | include file containing the .res files to return from the slave to the master |
| <i>fpvm3.h</i> | PVM include file |
| <i>get_result.f</i> | FORTTRAN source code which resides on the master and receives the results from the slaves |
| <i>Makefile</i> | make file to make a new PARJOB executable |
| <i>parjob_loc.f</i> | specifies the location of the <i>parjob.exe</i> executable on the hard disk |
| <i>run_job.f</i> | runs TPA analysis |
| <i>tpa_loc.f</i> | specifies the location of the <i>tpa.e</i> executable on the hard disk. (Can point to different <i>tpa.e</i> executables on different slaves) |
| <i>write_result.f</i> | reads the direct access files and writes text-based results files |

Installation of PARJOB requires a FORTRAN 77 compiler and a previous installation of PVM. Four changes are required to customize the PARJOB for installation and make a new PARJOB executable. These changes are described in the following sections.

1. Edit *parjob_loc.f* and specify the correct path of the PARJOB executable on your hard disk. For example, if the directory containing *parjob.exe* is */earth/user/nrc/parjob*, then *parjob_loc.f* is as follows

```

C*****
      subroutine parjob_loc (pjpath)
      implicit none
C
C      defines the variable pjpath, which is a string that describes the
C      pathname for the executable copy of PARJOB for use in spawning
C      copies of this process on the slaves, from the master process

```

C pjpath is a string containing the pathname for parjob.exe

character*80 pjpath

C*****

Cpjpath

 pjpath = '/earth/user/nrc/parjob/parjob.exe'

 return

 end

The path name assigned to pjpath is limited to 80 characters.

2. Edit *tpa_loc.f* and specify the correct path of the TPA executable on your hard disk. The variable *hostname* contains the name of the slave on which this routine is being executed. One can use *hostname* to execute different versions of TPA on different slaves. For example, if your directory containing *tpa.e* is */hornet/user/nrc/tpa* when on workstation *hornet* and */foxtrot/user/nrc/tpa* when on workstation *foxtrot*, then *tpa_loc.f* is as follows

C*****

 subroutine tpa_loc(tpapath,hostname)

 implicit none

C

C defines the variable tpapath, which is a string that describes the

C pathname for the executable copy of TPA for use in running

C TPA on the slaves.

C tpapath is a string containing the pathname for parjob.exe

 character*80 tpapath

 character*16 hostname

C*****

Ctpapath

C user must modify: hostname string

C pathname for tpa.e

C do not change: timex nice -5

C >&tpa.mode2

C if 'timex nice -5' is not included in the string entered into tpapath,

C then the runtime of tpa on the slaves will not be printed out in the

C tpa.mode2 log file and the niceness of running tpa.e on each slave will

C be set to default, which will take up more CPU and memory.

C if '>&tpa.mode2' is not included in the string tpapath, then

C no log file of the TPA run will be created; thus, in case of error,

C the user will be unable to see the error output for TPA on the slaves.

 if (hostname.eq.'foxtrot') then

 tpapath='timex nice -5 /foxtrot/user/nrc/tpa/tpa.e >&tpa.mode2'

 else if (hostname.eq.'hornet') then

 tpapath='timex nice -5 /hornet/user/nrc/tpa/tpa.e >&tpa.mode2'

 end if

 return

 end

The path name assigned to `tpath` is limited to 80 characters.

The “timex” command is used to record the CPU and wall clock time required for execution and is optional.

By default, TPA is executed with a “nice” value of 5. The user can modify the `tpath` variable to use another “nice” value.

By default, all TPA screen output is redirected into a file called `tpa.mode2`. The user can modify the `tpath` variable to redirect output to a different file.

3. Edit the “make” file to point to the correct locations of the PVM libraries, *libfpvm3.a* and *libpvm3.a*. These libraries are specified with the FLIBS variable in the “make” file and are in bold type in the example “make” file shown below.

```
FFLAGS      = -g
DEST        = .
HDRS        =
INSTALL     = /etc/install
LD          = f77
LDFLAGS     =
PVMDIR      = $(PVM_ROOT)
FLIBS       = -lsocket -lnsl \
              ./parjob.a \
              /bscr3/ccowan/pvm3/lib/SUN4SOL2/libfpvm3.a \
              /bscr3/ccowan/pvm3/lib/SUN4SOL2/libpvm3.a
MAKEFILE    = Makefile
OBJS        = get_result.o \
              run_job.o \
              write_result.o \
              parjob_loc.o \
              tpa_loc.o
PRINT       = pr
PROGRAM     = parjob.exe
SHELL       = /bin/sh
SRCS        = get_result.f \
              run_job.f \
              write_result.f \
              parjob_loc.f \
              tpa_loc.f
SYSHDRS     =
all: $(PROGRAM)
```

```
$(PROGRAM): $(OBJS) $(LIBS)
@echo "Linking $(PROGRAM) ..."
@$ (LD) $(FFLAGS) $(OBJS) $(FLIBS) -o $(PROGRAM)
@echo "done"
```

- The PVM_ROOT environment variable used in the definition of PVMDIR is set by the system administrator who installed the PVM system.
4. Type “make” at the command prompt to recompile any changed routines and make a new PARJOB executable.

4.2 UPGRADING TO FUTURE VERSIONS OF PVM

PARJOB has been tested with PVM versions 3.3 and 3.4beta6; however, PARJOB can be easily modified to work with other versions of PVM. The actions required are:

- Copy the *fpvm3.h* associated with the version of PVM you want to use to the directory containing the PARJOB source. Keep the name *fpvm3.h*.
- Modify the make file FLIBS variable to point to the *libfpvm3.a* and *libpvm3.a* libraries associated with the version of PVM you want to use.
- Type “touch tpa_loc.f” at the command line to force the make file to create a new executable.
- Type “make” at the command line to create a new executable.

5 RUNNING PARJOB WITH TPA

5.1 SETTING UP THE PARALLEL VIRTUAL MACHINE AND WORKING DIRECTORIES

The hosts to be included in the virtual machine and the working directories for each host are specified in a PVM host file. The host file is a text file that specifies which hosts to include in the virtual machine and the working directory for each host. For example, to create a virtual machine with four hosts, each with a unique working directory, the following PVM host file could be used. More information about PVM host files can be found on page 29, section 3.8 of the PVM manual (Geist et al., 1994).

```
* wd=/earth/users/ppir07/bigbend
bigbend
* wd=/earth/users/ppir07/foxtrot
foxtrot
* wd=/earth/users/ppir07/hornet
hornet
* wd=/earth/users/ppir07/goliath
goliath
```

For example, if the host file has been created with the file name "*hostfile*", PVM is invoked with the command

```
% pvm hostfile
```

At this point, the user should see a PVM prompt. The user should type "conf" to see the configuration of the virtual machine and ensure that it is configured correctly. Note that PVM will not read the host file if PVM is already running. The user must halt PVM and then invoke PVM with the host file. PVM is halted by issuing the "halt" command from within PVM. If the user wishes to change the hosts in the virtual machine, PVM must be halted and restarted with a new host file. Type "pvm" to enter the pvm console. Type "halt" to stop pvm and exit.

The user must specify a unique working directory for each slave process. This is so that there will be no input and output file contentions between slave processes running TPA. PARJOB verifies that each slave has a unique working directory. The local */tmp* directory may be used as a working directory. Use of a local directory may speed up the analysis. For example, the user creates a directory in */tmp* on the local hard disk of four workstations hornet, foxtrot, bigbend, and goliath. The PVM host file would look like

```
* wd=/tmp/hornet
hornet
* wd=/tmp/foxtrot
foxtrot
* wd=/tmp/bigbend
bigbend
* wd=/tmp/goliath
goliath
```

In each working directory, the user must copy the unmodified *tpa.inp* input file. This can most easily be done with a script file to reduce the amount of keyboard command entry. Thus, the master directory and all slave working directories must contain the identical *tpa.inp* input file before the analysis begins. An example of a script file is as follows:

```
#
mkdir /tmp/bigbend
rsh hornet "mkdir /tmp/hornet"
rsh foxtrot "mkdir /tmp/foxtrot"
rsh sisyphus "mkdir /tmp/sisyphus"
rsh goliath "mkdir /tmp/goliath"
cp tpa.inp /tmp/bigbend
rsh hornet "cp pvm/parjob2/tpa.inp /tmp/hornet"
rsh foxtrot "cp pvm/parjob2/tpa.inp /tmp/foxtrot"
rsh sisyphus "cp pvm/parjob2/tpa.inp /tmp/sisyphus"
rsh goliath "cp pvm/parjob2/tpa.inp /tmp/goliath"
echo done
```

If execution is terminated prematurely, one must copy an unmodified version of *tpa.inp* again to all slave working directories before re-executing the code.

5.2 INVOKING PARJOB

PARJOB can be invoked from the command line by typing the executable name *parjob.exe*. This will invoke the PARJOB master process which will in turn spawn the slave processes. The master process always runs on the host on which it is invoked. It is useful to invoke PARJOB with a “nice” command as a courtesy to other users and to prevent conflicts if a slave process is running on the same host as the master. The master spends the majority of its time waiting for results from the slaves and does not require a high priority to run effectively. For example, to invoke PARJOB with a nice value of 5 type

```
nice -5 parjob.exe
```

5.3 SETTING THE NUMBER OF SLAVE PROCESSES

PARJOB currently lets PVM decide the order in which processes are allocated to the slaves. For example, if there are six hosts in the connected cluster comprising the virtual machine called Earth, Mercury, Venus, Mars, Saturn, and Jupiter, and four slave processes are requested, then PVM will typically assign slave processes on the first four hosts in the pvm hostfile; the remaining hosts will be idle. PVM will not assign two slave processes on a single host if there are free hosts available. If you specify N slaves with N hosts, PVM will assign one slave on each host. Thus, one host will have the master and one of the slave processes. If more than N slaves are specified with only N hosts, PVM will assign more than one slave to a host. The user is cautioned that this will not work correctly with this version of PARJOB, even if the host has multiple processors, because each process on a host will have the same working directory.

5.4 HALTING PVM

PVM is halted by typing “halt” at the command line from within PVM. This is necessary if the user wants to change the hosts in the virtual machine. It is also useful if there is a problem running PVM. Halting PVM will often clean up extraneous files and processes.

5.5 SUMMARY FILE

PARJOB creates a log file that contains a summary of the analysis and all transactions between the master and slave. The log file will be named “*jobname.log*”, where *jobname* is entered by the user during PARJOB execution. A summary is also written to the screen so that the user can assess the state of the analysis.

6 RESTRICTIONS

6.1 HOMOGENEOUS SYSTEMS

PARJOB was designed for a heterogeneous system but to date has only been tested on a homogeneous system of either HP, Sun, or SGI workstations. Modifications for heterogeneous systems should be straightforward.

6.2 .rhosts FILE

The *.rhosts* file (found in the user's home directory) contains a list of all workstations the user can access **without a password**. Alternatively, the system root directory may contain a *.rhosts* file that allows a user access to another workstation without a password. For example, if the user wishes to use workstations Earth, Mars, and Jupiter, the *.rhosts* file for his account must contain these machine names.

If the *.rhosts* file is not modified, the user must include the user name and password in the PVM host file, see page 29, section 3.8 of the PVM manual (Geist et al., 1994).

6.3 HARD DISK ACCESS

PARJOB assumes that the CPUs on which the slaves run all have access to the hard disk where *parjob.exe* is located. For example, if the PARJOB executable is located on hard disk */earth/data/parjob* and slave processes are run on Mars and Saturn, Mars and Saturn must be able to access the */earth* hard disk.

6.4 UNIQUE WORKING DIRECTORY

It is incumbent upon the user to have a unique working directory established for each slave process used in an analysis. The working directories are specified in the PVM host file. PARJOB will detect the situation where multiple slaves have the same working directory, warn the user, and terminate. If the user specifies a working directory in the PVM host file that does not exist, the slave process will use the user's home directory.

6.5 NUMBER OF SLAVE PROCESSES

PARJOB currently has a limit of 20 slave processes. This can be expanded if needed.

6.6 ANALYZE ALL VECTORS

PARJOB-TPA can only solve for all specified vectors, (NumberOfRealizations in the *tpa.inp* file). For example, it cannot solve for vectors 3 through 5 out of 10 vectors. If a subset of vectors is specified, PARJOB will ignore the subset and analyze all 10 vectors.

7 TROUBLESHOOTING

7.1 READING SLAVE RESULTS

PVM writes the PARJOB related standard output from all slaves to a file named */tmp/pvml.uid*, on the local temp directory of the host on which *parjob.exe* was invoked, where uid is an integer unique to each user.

All TPA output on the slaves is written to a file "tpa.mode2" contained in each working directory.

7.2 ERROR RUNNING PVM

Two simple approaches can be used to eliminate the error condition. The first is to try halting PVM then restarting. To do this, type “pvm” to enter the PVM console. Type “halt” to exit PVM.

The second approach is to delete the *PVM** files located on the temporary disk of each host. Try deleting the *pvm** and *pvm** files on the */tmp* directory of the host. This is accomplished easily using a remote shell command. For example, to delete file */tmp/pvm.203* on host Saturn, use the command

```
remsh saturn 'rm /tmp/pvm*'.
```

7.3 CAN'T ADD HOST

Try deleting the *pvm** and *pvm** files on the */tmp* directory of the host. This is accomplished easily using a remote shell command. For example, to delete file */tmp/pvm.203* on host Saturn, use the command

```
remsh saturn 'rm /tmp/pvm*'.
```

7.4 TPA ABORTS ON SLAVE

TPA may abort on the slave when TPA is being run in conjunction with other large programs on the same machine. There is no known general solution to this. The file *tpa.mode2* can be queried to determine the cause of failure. Often, TPA will run correctly on the slave at a later time.

8 VERIFICATION

A number of verification problems have been run using PARJOB-TPA and compared with a standard sequential analysis using only TPA. In all cases, the results files are identical. Table 3 presents the test matrix used for in code verification. The first two columns present the *tpa.inp* parameters and flags that were varied for the test cases. Other columns represent the variations from the base case. The headings for these columns are the names of input files and uniquely identify the test cases. For each case, only the differences from the base case input file are listed.

9 REFERENCES

- Millwater, H., J. Wu, B. Thacker, and R. Janetzke. 1998. *Efficient Uncertainty and Sensitivity Analysis Methods for Performance Assessment of Complex Systems*. SwRI Project 06-9994. San Antonio, TX: Southwest Research Institute.
- Geist, A., A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. 1994. *PVM: Parallel Virtual Machine, A User's Guide and Tutorial for Networked Parallel Computing*. Cambridge, Massachusetts,: MIT Press.
- Mohanty, S., and T.J. McCartin. 1998. *Total-system Performance Assessment (TPA) Version 3.2 Code: Module Descriptions and User's Guide*. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

Table 3. Test case parameter values

| tpa.iup PARAMETERS | TEST CASE | | | | | | | | | | | | | | | | | | |
|--|-----------------|----------------------------|----------------------------|---------------------|-----------------------|--------------------------|-----------------------------|------------------|------------------|--------------------|--------------------|---------------------|---------------------|--------------------|---------------------------|---------------------------|-----------------------|-----------------------|------------------------------|
| | <i>base.inp</i> | <i>scenflag1.drel1.inp</i> | <i>scenflag1.drel0.inp</i> | <i>rectzone.inp</i> | <i>allsubarea.inp</i> | <i>CheckNucerror.inp</i> | <i>1nuclidein1chain.inp</i> | <i>2real.inp</i> | <i>1real.inp</i> | <i>100real.inp</i> | <i>500real.inp</i> | <i>chgtime1.inp</i> | <i>chgtime2.inp</i> | <i>RHFlag1.inp</i> | <i>FlowModelflag1.inp</i> | <i>InvertBypFlag1.inp</i> | <i>RefluxMod2.inp</i> | <i>RefluxMod1.inp</i> | <i>StopAtSubSubarea1.inp</i> |
| VolcanismDisruptiveScenarioFlag (0,1) | 0 | 1 | 1 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| FaultingDisruptiveScenarioFlag (0,1) | 0 | 1 | 1 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| SeismicDisruptiveScenarioFlag (0,1) | 1 | 0 | 0 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Subareas (one rect. zone, seven subareas) | seven sub-areas | .. | .. | one rect. zone | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| StopAtSubarea (0 or 7,1) | 3 | .. | .. | 0 | 7 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 1 |
| CheckNuclidesAndChains (0, 1) n.b. 1 should give error in TPA run on each slave | 0 | .. | .. | .. | .. | 1 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Number of Nuclides w/i each Chain (1,20) | 20 | .. | .. | .. | .. | .. | 1 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| Number of Chains (1,13) | 13 | .. | .. | .. | .. | .. | 1 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| DirectReleaseOnlyFlag (0, 1) n.b. change only if V.D.S.Flag is 1 | 0 | 1 | 0 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| NumberOfRealizations (1,2,10,100,500) | 10 | .. | .. | .. | .. | .. | .. | 2 | 1 | 100 | 500 | .. | .. | .. | .. | .. | .. | .. | .. |

Table 3. Test case parameter values (cont'd)

| tpa.iup PARAMETERS | TEST CASE | | | | | | | | | | | | | | | | | | |
|--|-----------------|----------------------------|----------------------------|---------------------|-----------------------|--------------------------|------------------------------|------------------|------------------|--------------------|--------------------|----------------------|----------------------|---------------------|---------------------------|---------------------------|-----------------------|-----------------------|------------------------------|
| | <i>base.inp</i> | <i>scenflag1.drel1.inp</i> | <i>scenflag1.drel0.inp</i> | <i>rectzone.inp</i> | <i>allsubarea.inp</i> | <i>CheckNucerror.inp</i> | <i>1nucleidein1chain.inp</i> | <i>2real.inp</i> | <i>1real.inp</i> | <i>100real.inp</i> | <i>500real.inp</i> | <i>cbngtime1.inp</i> | <i>cbngtime2.inp</i> | <i>RHFflag1.inp</i> | <i>FlowModelflag1.inp</i> | <i>InvertBypFlag1.inp</i> | <i>RefluxMod2.inp</i> | <i>RefluxMod1.inp</i> | <i>StopAtSubSubarea1.inp</i> |
| DurationOfCompliancePeriod (1.0E4,2.0E4) | 1.E4 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 2.E4 | 2.E4 | .. | .. | .. | .. | .. | .. |
| MaximumTime (1.0E4,2.0E4,1.0E5) | 1.E4 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 2.E4 | 1.E5 | .. | .. | .. | .. | .. | .. |
| NumberOfTimeStepsInCompliancePeriod (21, 101, 201) | 201 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 101 | .. | .. | .. | .. | .. | .. |
| NumberOfTimeStepsAfterCompliancePeriod (20, 100) | 0 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 100 | .. | .. | .. | .. | .. | .. |
| TabularTemperatureRHFlag (0,1) | 0 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 1 | .. | .. | .. | .. | .. |
| SelectRefluxModel (1, 2, 3) n.b. changes results files | 3 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 2 | 1 | .. |
| TimeOfBackfillEmplaced (100001, 100) n.b. changes results files | 100001 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 100 | 100 | .. | .. | .. | .. | .. | .. |
| FlowModelFlag (0,1) | 0 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 1 | .. | .. | .. | .. |
| InvertBypass (0,1) | 0 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 1 | .. | .. | .. |
| DistanceToReceptorGroup (10, 20) n.b. use 20 for larger values of MaxTime or CompliancePeriod | 10 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | 20 | 20 | .. | .. | .. | .. | .. | .. |

41/63

PARJOB, Parallel Processing Program for TPA Version 3.2

User's Manual

**Harry Millwater, Nathan Kelley
Probabilistic Mechanics and Reliability Section
Southwest Research Institute**

November 20, 1998

for Center for Nuclear Waste Regulatory Analysis

1.1 Introduction

The Total Performance Assessment (TPA) computer code is used to assess the performance of a geologic repository in terms of the amount of radionuclide release. TPA is capable of performing uncertainty analysis by considering many of the input parameters as random variables. The methodology used to determine the complementary cumulative distribution function (CCDF) of the result is Latin Hypercube Sample (LHS). LHS involves repeated calculations of radionuclide release subject to a number of sets of different input parameters, called "vectors". The computational time to develop a CCDF may be prohibitive if a large number of vectors is required, the solution to each vector is time consuming, and the solutions are computed serially, i.e., one after another on a single processor. Parallel processing methodology, on the other hand, has the potential for significantly reducing the computational time to develop the CCDF with no loss in accuracy by computing the solutions to each vector in parallel. Problems which were previously prohibitive may become practical if solved using a parallel processing methodology.

Southwest Research Institute Probabilistic Mechanics Section has recently completed a focused internal research project entitled "Efficient Uncertainty and Sensitivity Analysis Methods for Performance Assessment of Complex Systems." One task in this multi-objective project was to implement parallel processing methodology to reduce computation time for probabilistic analysis. A probabilistic methodology and software were developed as an outcome of the project. The software, PARJOB - PARallel processing of JOBs, was demonstrated on an HP workstation system using the TPA 2.0 computer program. PARJOB was very effective in reducing the computational time for computing the LHS vectors. The table below shows the speedup obtained for a 1000 vector analysis using 11 workstations compared to a single (but fastest) machine, i.e., the sequential solution was run on the fastest HP machine. The solution time was reduced from approximately 31 hours to 6 hours.

| Analysis Approach | Solution Time (Wall Time) |
|-----------------------------------|---------------------------|
| Single machine (lad) | 30 hours 45 minutes |
| Parallel Processing (11 machines) | 5 hours 44 minutes |

**Comparison of Solution Times for Parallel and non-Parallel 1000 Vector TPA Analysis
(Speedup = 5.36)**

PARJOB is written entirely in Fortran 77 and PVM (Parallel Virtual Machine). PVM is free public domain software from Oakridge National Labs and is used for all message passing. PVM is available on most all Unix operating systems. Thus, PARJOB is very transportable. It has been tested successfully on HP, Sun and Silicon Graphics workstations.

PARJOB has been designed with a number of fault tolerant features. A summary is given below.

- detect when a Slave or Master process or host dies,
- detect an incorrect working directory,
- detect when a job aborts,
- place a time limit on analysis,
- rerun a job which has failed once, but not twice,
- record results as soon as completed,
- automatic restart from a partially completed analysis.

Summary of Fault Tolerant Features

Because of the success of the parallel processing methodology using TPA 2.0, it was decided by the Center for Nuclear Waste Regulatory Analysis (CNWRA) to explore the application of PARJOB to TPA version 3. The resulting computer code and its use is described in this user's manual.

1.2 Overview

PARJOB uses a Master-Slave paradigm for probabilistic analysis. This approach is ideal for LHS because the same analysis model is executed with different inputs. Each analysis of a vector is performed in a Slave process and does not need to communicate with other Slaves. The vector inputs are sent from the Master to the Slave and the response is sent from the Slave back to the Master.

The PARJOB-TPA flowchart, shown on the next page, gives an overview of the procedure. Only minor modifications to TPA version 3 were required to interface with PARJOB. All changes to TPA were performed by CNWRA staff. TPA was modified such that its analysis can be performed in three modes:

- Mode 1: create LHS sample file - called "lhs.out",
- Mode 2: compute results for each vector in the LHS - on Slave,
- Mode 3: create ccdf related results files once all other results files have been completed.

Modes 1 and 3 are performed by TPA running in the Master process. Mode 2 is performed one vector at a time by the Slave processes and the results sent to the Master.

1.2.1 Generating LHS samples (Mode 1)

PARJOB internal subroutine pvmprep modifies the standard tpa.inp input file to tell TPA to run in mode 1. TPA is run to generate the lhs vectors in a file called "lhs.out". TPA then returns to PARJOB.

The changes to tpa.inp are:

```
NumberOfRealization = total no. of vectors
Start at Realization = 1
Stop at Realization = 1
```

Inserted at the end of tpa.inp:

```
**
IFLAG
PVMPREPROCESSOR(Yes=1, No = 0)
1
EndOfFile
```

1.2.2 Running the Analyses (Mode 2)

On the Slave process, pvmprep is called to modify the tpa.inp file to tell TPA to run in mode 2 and to analyze only 1 vector. The particular vector to analyze is passed to the Slave by the Master. The Slave writes a local lhs.out file which contains the vector passed from the Master to the Slave. TPA runs the single vector input and generates the .res output files with the exception that ccdf-related .res files are not created. ccdf-related files require results from all vectors.

The changes to tpa.inp are:

```
NumberOfRealization = vector #
Start at Realization = vector #
Stop at Realization = vector #
```

where vector # is the number of the vector to be analyzed.

Inserted at the end of tpa.inp:

```
**
IFLAG
PVM (Yes=1, No = 0)
1
EndOfFile
```

1.2.3 Returning and Storing the Results

The results from the single-vector TPA analyses are returned to the Master for storage and ultimately, assembly into the final results files. PARJOB has been designed so that it does not need to know the format of the results files. This is very significant because there are 20 results files that must be read and returned. In addition, no changes to PARJOB are required should the format of the results files be modified. The filenames to return to the Master are contained in file "filenames.i".

On the Slave, PARJOB reads each line in the results file into a character variable. The number of lines and the character variables are sent back to the Master for each .res file contained in filenames.i. The Master unpacks the data and stores the results into a direct access file. The direct access files are used internally and are titled according to its corresponding file, e.g., cumrel.da.

A list of the results files returned to the Master is

- airpkdos.res
- airpkdos_c.res
- ashout.res
- cumrel.res
- cumrel_c.res
- gwpkdos.res
- gwpkdos_c.res
- gwtuzsz.res
- infilper.res
- nearfld.res
- npkdoset.res
- npkdoset_c.res
- pkreltim.res
- pkreltim_c.res
- relgwgs.res
- relgwgs_c.res
- samplpar.res
- totdose.res
- totdose_c.res
- wpsfail.res

1.2.4 Assembling the Results Files

Once all vectors have been analyzed, the Master process reads the direct access file corresponding to each result file, e.g., cumrel.da, and reconstructs the cumulative results file for all vectors, e.g., cumrel.res.

1.2.5 Creating the CCDF Related Results Files (Mode 3)

Once the non-CCDF related results files are created, TPA is run in mode 3 in order to create the CCDF related results file, e.g., gsccdf.res. Subroutine pvmprep is called first to modify the tpa.inp input file for mode 3, then TPA is called. TPA will read the existing *.res files and generate the non-CCDF related results files.

A list of the ccdf-related files generated by TPA in mode 3 is

- gsccdf.res
- gsccdf_c.res
- gwccdf.res
- gwccdf_c.res
- relccdf.res
- relccdf_c.res

The changes to tpa.inp are:


```

NumberOfRealization = 0
Start at Realization = 1
Stop at Realization = 1

```

Inserted at the end of tpa.inp:

```

**
IFLAG
PVMPOSTPROCESSOR (Yes=1, No = 0)
1
EndOfFile

```

1.3 Installation and Customization

1.3.1 PARJOB

PARJOB can be installed into any directory of convenience. A list of the files on the PARJOB distribution is given below. The user should only have to modify *parjob_loc.f* and *tpa_loc.f* as described below.

| File | Description |
|---------------------|---|
| parjob.a | library containing the core PARJOB routines |
| filenames.i | text file containing the .res files to return from the Slave to the Master |
| fpvm3.h | PVM include file |
| get_result.f | Fortran source code which resides on the Master and receives the results from the Slaves |
| Makefile | make file to make a new PARJOB executable |
| <i>parjob_loc.f</i> | specifies the location of the parjob.exe executable on the hard disk |
| run_job.f | runs TPA analysis |
| <i>tpa_loc.f</i> | specifies the location of the tpa.e executable on the hard disk. (Can point to different tpa.e executables on different slaves) |
| write_result.f | reads the direct access files and writes text-based results files |

Installation of PARJOB requires a Fortran 77 compiler and a previous installation of PVM. Four changes are required to customize the PARJOB for your installation and make a new PARJOB executable.

- 1) Edit *parjob_loc.f* and specify the correct path of the PARJOB executable on your hard disk. For example, if the directory containing *parjob.exe* is */earth/user/nrc/parjob*, then, *parjob_loc.f* would look like

```

C*****
      subroutine parjob_loc (pjpath)
      implicit none
C
C      defines the variable pjpath, which is a string that describes the
C      pathname for the executable copy of PARJOB for use in spawning
C      copies of this process on the slaves, from the master process
C
C      PJpath is a string containing the pathname of for parjob.exe
      character*80 pjpath
C*****
Cpjpath

```

```

    pjpath = '/earth/user/nrc/parjob/parjob.exe'

    return
end

```

The path name is limited to 80 characters.

2) Edit tpa_loc.f and specify the correct path of the tpa executable on your hard disk. The variable hostname contains the name of the slave on which this routine is being executed. One can use hostname to execute different versions of tpa on different slaves. For example, if your directory containing tpa.e is /hornet/user/nrc/tpa when on workstation hornet and /foxtrot/user/nrc/tpa when on workstation foxtrot, then, tpa_loc.f would look like

```

C*****
      subroutine tpa_loc(tpapath,hostname)
      implicit none

C
C      defines the variable tpapath, which is a string that describes the
C      pathname for the executable copy of TPA for use in running
C      TPA on the slaves.

C      tpapath is a string containing the pathname of for parjob.exe

      character*80 tpapath
      character*16 hostname

C*****

Ctpapath

C      user must modify: hostname string
C                        pathname for tpa.e
C      do not change:  timex nice -5
C                        >&tpa.mode2
C      if 'timex nice -5' is not included in the string entered into tpapath,
C      then the runtime of tpa on the slaves will not be printed out in the
C      tpa.mode2 log file and the niceness of running tpa.e on each slave will
C      be set to default, which will take up more CPU and memory.
C      if '>&tpa.mode2' is not included in the string tpapath, then
C      no log file of the TPA run will be created; thus, in case of error,
C      the user will be unable to see the error output for TPA on the slaves.

      if (hostname.eq.'foxtrot') then
         tpapath='timex nice -5 /foxtrot/user/nrc/tpa/tpa.e >&tpa.mode2'
      else if (hostname.eq.'hornet') then
         tpapath='timex nice -5 /hornet/user/nrc/tpa/tpa.e >&tpa.mode2'
      end if

      return
end

```

The path name is limited to 80 characters.

- The timex command is used to record the CPU and wall time required by the analysis and is optional.
- By default, tpa is executed with a nice value of 5. The user can modify the tpapath variable to use another nice value.
- By default, all tpa screen output is redirected into a file called tpa.mode2. The user can modify the tpapath variable to redirect output to different file.

- 3) Edit the make file to point to the correct locations of the PVM libraries, `libfpvm3.a` and `libpvm3.a`. These libraries are specified with the `FLIBS` variable in the make file and are in bold type in the example makefile shown below.

```

FFLAGS      = -g
DEST         = .
EXTHDRS      = /bscr3/ccowan/pvm3/include/fpvm3.h
HDRS         =
INSTALL      = /etc/install
LD           = f77
LDFLAGS      =
PVMDIR       = $(PVM_ROOT)
FLIBS        = -lsocket -lnsl \
               ./parjob.a \
               /bscr3/ccowan/pvm3/lib/SUN4SOL2/libfpvm3.a \
               /bscr3/ccowan/pvm3/lib/SUN4SOL2/libpvm3.a
MAKEFILE     = Makefile
OBJS         = get_result.o \
               run_job.o \
               write_result.o \
               parjob_loc.o \
               tpa_loc.o
PRINT        = pr
PROGRAM      = parjob.exe
SHELL        = /bin/sh
SRCS         = get_result.f \
               run_job.f \
               write_result.f \
               parjob_loc.f \
               tpa_loc.f
SYSHDRS      =
all:          $(PROGRAM)
$(PROGRAM):  $(OBJS) $(LIBS)
              @echo "Linking $(PROGRAM) ..."
              @$ (LD) $(FFLAGS) $(OBJS) $(FLIBS) -o $(PROGRAM)
              @echo "done"

```

- 4) Type “make” at the command prompt to recompile any changed routines and make a new `parjob` executable.

1.3.2 Upgrading to Future Versions of PVM

PARJOB has been with tested with PVM versions 3.3 and 3.4beta6; however, PARJOB can be easily modified to work with other versions of PVM. The actions required are:

- 1) Copy the `fpvm3.h` associated with the version of PVM you want to use to the directory containing the PARJOB source. Keep the name `fpvm3.h`.
- 2) Modify the make file `FLIBS` variable to point to the `libfpvm3.a` and `libpvm3.a` libraries associated with the version of PVM you want to use.

- 3) Type “touch tpa_loc.f” at the command line to force the make file to create a new executable in step 4.
- 4) Type “make” at the command line to create a new executable.

1.4 Running PARJOB-TPA

1.4.1 Setting up the Parallel Virtual Machine and Working Directories

The hosts to include in the virtual machine and the working directories for each host are specified in a PVM host file. The host file is a text file which specifies which hosts to include in the virtual machine and the working directory for each host. For example, to create a virtual machine with 4 hosts, each with a unique working directory, the following PVM host file could be used. More information about PVM host files can be found on page 29, section 3.8 of the PVM manual.²

```
* wd=/earth/users/ppir07/bigbend
bigbend
* wd=/earth/users/ppir07/foxtrot
foxtrot
* wd=/earth/users/ppir07/hornet
hornet
* wd=/earth/users/ppir07/goliath
goliath
```

If the user wishes to change the hosts in the virtual machine, PVM must be halted and restarted with a new host file. Type “pvm” to enter the pvm console. Type “halt” to stop pvm and exit. Type “pvm hostfile_name” to start pvm with the options specified in the host file.

For example, a PVM host file has been created with the file name “hostfile”. PVM is invoked with the command

```
pvm hostfile
```

At this point, the user should see a PVM prompt. The user should type “conf” to see the configuration of the virtual machine and ensure that it is configured correctly. Note, PVM will not take the host file if it is already running. The user must halt pvm then invoke pvm with the host file. PVM is halted by issuing the “halt” command from within PVM.

The user must specify a unique working directory for each slave process. This is so that there will be no input and output file contentions between slave processes running tpa. PARJOB verifies that each slave has a unique working directory. The local /tmp directory may be used as a working directory. This may speed up the analysis. For example, the user creates a directory on the local hard disk of workstation hornet. The pvm host file would like

```
* wd=/tmp/bigbend
bigbend
* wd=/tmp/foxtrot
foxtrot
* wd=/tmp/hornet
hornet
* wd=/tmp/goliath
goliath
```

In each working directory, the user must copy the unmodified tpa.inp input file. Thus, the master directory and all slave working directories must contain the identical tpa.inp input file before the analysis begins.

If the analysis gets terminated prematurely, copy an unmodified version of tpa.inp again to all slave working directories before beginning another analysis.

1.4.2 Invoking PARJOB

PARJOB can be invoked from the command line by typing the executable name parjob.exe. This will invoke the PARJOB Master process which will in turn spawn the Slave processes. The Master process always runs on the host on which it is invoked. It is useful to invoke parjob with a “nice” command. This is nice for other users and useful if a Slave process is running on the same host as the Master. The Master spends the majority of its time waiting for results from the Slaves and does not require a high priority to run effectively. For example, to invoke the Slave with a nice value of 5 type

```
nice -5 parjob.exe
```

1.4.3 Setting the Number of Slave Processes

PARJOB currently lets PVM decide which processes to locate on which Slaves. For example, if you have 6 hosts in your virtual machine: Earth, Mercury, Venus, Mars, Saturn and Jupiter and spawn 4 Slave processes, PVM will typically put Slave processes on the first 4 hosts in the pvm hostfile; the remaining hosts will be idle. Mercury, Venus, Mars and Saturn, although this is not guaranteed. It will not put two Slave processes on a single host if there are free hosts available. If you specify N Slaves with N hosts, PVM will put a Slave on each host. Thus, one host will have the Master and a Slave process. If you specified more than N Slaves with only N hosts, PVM will put more than one Slave on a host. This will not work correctly with this version of PARJOB, even if the host has multiple processors, because each process on a host will have the same working directory.

1.4.4 Halting PVM

PVM is halted by typing “halt” at the command line from within PVM. This is necessary if the user wants to change the hosts in the virtual machine. It is also useful if there is a problem running PVM. Halting PVM will often clean up extraneous files and processes.

1.4.5 Summary File

PARJOB creates a log file which contains a summary of the analysis and all transactions between the Master and Slave. The user is prompted for a file name. The log file will be named “jobname.log”, where jobname is the name entered by the user.

A summary is also written to the screen so that the user can assess the state of the analysis.

1.5 Restrictions

1.5.1 Homogeneous Systems

PARJOB was designed for a heterogeneous system but has currently only been tested on a homogeneous system, either, HP, Sun, or SGI. Modifications for heterogeneous systems should be straightforward.

1.5.2 .rhosts File

The .rhosts file (found in the user’s home directory) contains a list of all workstations the user can access *without a password*. Alternatively, the root directory may contain a .rhosts file which allows a user access to another workstation without a password. For example, if the user wishes to use workstations earth, mars, and jupiter, the .rhosts file for his account must contain these machines.

If the .rhosts file is not modified, the user must include the user name and password in the pvm host file, see page 29, section 3.8 of the PVM manual.²

1.5.3 Hard Disk Access

PARJOB assumes that the CPUs on which the Slaves run can all have access to the hard disk where parjob.exe is located. For example, if the PARJOB executable is located on hard disk

/earth/data/parjob and Slave processes are run on Mars and Saturn, Mars and Saturn must be able to access the /earth hard disk.

1.5.4 Unique Working Directory

It is incumbent upon the user to have a unique working directory established for each Slave process used in an analysis. The working directories are specified in the PVM host file. PARJOB will detect the situation where multiple Slaves have the same working directory, will warn the user, and terminate. If the user specifies a working directory in the PVM host file which does not exist, the Slave process will use the user's home directory.

1.5.5 Number of Slave Processes

PARJOB currently has a limit of 20 slave processes. This can be expanded if needed.

1.5.6 Analyze All Vectors

PARJOB-TPA can only solve for ALL specified vectors, (NumberOfRealizations in the tpa.inp file). That is, for example, it cannot solve for vectors 3 through 5 out of 10 vectors. If a subset of vectors is specified, PARJOB will ignore the data and analyze all 10 vectors.

1.6 Verification

A number of verification problems have been run using PARJOB/TPA and compared with a standard sequential analysis using only TPA. In all cases, the results files are identical. A table describing the test matrix is given below. For each case, only the differences from the base case are listed.

5/63

VERIFICATION PROCESS FOR PARJOB—INPUT DECKS AND PROCESSING CONDITIONS

| FILENAMES ----> | base .inp | scenflag1 drel1.inp | scenflag1 drel0.inp | rectzone .inp | allsubarea .inp |
|---|---|---|--------------------------------|--------------------------------|---------------------------------|
| PARAMETERS V | | | | | |
| VolcanismDisruptiveScenarioFlag (0,1) | 0 | 1 | 1 | .. | .. |
| FaultingDisruptiveScenarioFlag (0,1) | 0 | 1 | 1 | .. | .. |
| SeismicDisruptiveScenarioFlag (0,1) | 1 | 0 | 0 | .. | .. |
| Subareas (one rect. zone, seven subareas) | seven subareas | .. | .. | one rect. zone | .. |
| StopAtSubarea (0 or 7,1) | 3 | .. | .. | 0 | 7 |
| CheckNuclidesAndChains (0, 1) n.b. 1 should give error in tpa run on each slave | 0 | .. | .. | .. | .. |
| Number of Nuclides w/i each Chain (1,20) | 20 | .. | .. | .. | .. |
| Number of Chains (1,13) | 13 | .. | .. | .. | .. |
| DirectReleaseOnlyFlag (0, 1) n.b. change only if V.D.S.Flag is 1 | 0 | 1 | 0 | .. | .. |
| NumberOfRealizations (1,2,10,100,500) | 10 | .. | .. | .. | .. |
| DurationOfCompliancePeriod (1.0E4,2.0E4) | 1.00E+04 | .. | .. | .. | .. |
| MaximumTime (1.0E4,2.0E4,1.0E5) | 1.00E+04 | .. | .. | .. | .. |
| NumberOfTimeStepsInCompliancePeriod (21, 101, 201) | 201 | .. | .. | .. | .. |
| NumberOfTimeStepsAfterCompliancePeriod (20, 100) | 0 | .. | .. | .. | .. |
| TabularTemperatureRHFlag (0,1) | 0 | .. | .. | .. | .. |
| SelectRefluxModel (1, 2, 3) n.b. changes results files | 3 | .. | .. | .. | .. |
| TimeOfBackfillEmplaced (100001, 100) n.b. changes results files | 100001 | .. | .. | .. | .. |
| FlowModelFlag (0,1) | 0 | .. | .. | .. | .. |
| InvertBypass (0,1) | 0 | .. | .. | .. | .. |
| DistanceToReceptorGroup (10, 20) n.b. use 20 for larger values of MaxTime or CompliancePeriod | 10 | .. | .. | .. | .. |
| PROCESSING CONDITIONS V | | | | | |
| Name of Master | Scratchy1 | Bigbend | .. | .. | .. |
| Niceness of Master Processes | 30 | .. | .. | .. | .. |
| Number of Slaves Used (# processes) | 3 | .. | .. | .. | .. |
| Names of Slaves Used: # Users | Bigbend: 6 Scratchy1: 1 Hornet: 0 | Bigbend: 6 Scratchy1: 1 Hornet: 0 | | | |
| Niceness of Running TPA on Slaves | 25 | .. | .. | .. | .. |
| Average Runtime of TPA on each Slave | BB: 1:30 S: 4:30 H: 3:00 | BB: 0:40 S: 1:00 H: 0:30 | BB: 2:00 S: 6:00 H: 3:30 | BB: 1:00 S: 1:40 H: 1:20 | BB: 2:40 S: 11:00 H: 5:30 |
| Number of Jobs Per Slave | BB: 5 S: 2 H: 3 | BB: 3 S: 2 H: 5 | | BB: 4 S: 3 H: 3 | |
| Starting Time for Parjob Run (hr:min:sec) | 14:14:20 | 8:18:13 | 23:20:37 | 13:28:22 | 21:17:27 |
| Stopping Time for Parjob Run (hr:min:sec) | 14:25:50 | 8:22:05 | 23:33:52 | 13:34:48 | 21:40:58 |
| Parjob Runtime (hr:min:sec.fraction of sec) | 11:29.8 | 03:52.3 | 13:14.4 | 06:15.7 | 23:31.1 |
| Date of Parjob Run | 9/22/98 | 9/18/98 | 9/17/98 | 9/17/98 | 9/17/98 |
| Machine Used to Run TPA seq. | Scratchy1 | .. | .. | .. | .. |
| Niceness of Running TPA seq. | 25 | .. | .. | .. | .. |
| Number of Users | 2 | .. | .. | .. | .. |
| Starting Time for TPA seq. Run (hr:min:sec) | 12:09:34 | 17:51:54 | 17:02:16 | 16:46:03 | 10:30:48 |
| Stopping Time for TPA seq. Run (hr:min:sec) | 12:53:05 | 17:57:54 | 17:51:48 | 17:02:10 | 12:09:28 |
| TPA seq. Runtime (hr:min:sec.fraction of sec) | 43:30.8 | 05:59.6 | 49:32.3 | 16:07.3 | 1:38:37 |
| Date of Sequential Run | 9/27/98 | 9/27/98 | 9/27/98 | 9/27/98 | 9/27/98 |

VERIFICATION PROCESS FOR PARJOB--INPUT DECKS AND PROCESSING CONDITIONS

| FILENAMES -----> | CheckNuc error.inp | 1nuclidein 1chain.inp | 2real .inp | 1real .inp | 100real .inp |
|---|-----------------------|--------------------------|---------------|---------------|-----------------|
| PARAMETERS V | | | | | |
| VolcanismDisruptiveScenarioFlag (0,1) | .. | .. | .. | .. | .. |
| FaultingDisruptiveScenarioFlag (0,1) | .. | .. | .. | .. | .. |
| SeismicDisruptiveScenarioFlag (0,1) | .. | .. | .. | .. | .. |
| Subareas (one rect. zone, several subareas) | .. | .. | .. | .. | .. |
| StopAtSubarea (0 or 7,1) | .. | .. | .. | .. | .. |
| CheckNuclidesAndChains (0, 1) n.b. 1 should give error in tpa run on each slave | 1 | .. | .. | .. | .. |
| Number of Nuclides w/i each Chain | .. | 1 | .. | .. | .. |
| Number of Chains | .. | 1 | .. | .. | .. |
| DirectReleaseOnlyFlag (0, 1) n.b. change only if V.D.S.Flag is 1 | .. | .. | .. | .. | .. |
| NumberOfRealizations | .. | .. | 2 | 1 | 100 |
| DurationOfCompliancePeriod | .. | .. | .. | .. | .. |
| MaximumTime | .. | .. | .. | .. | .. |
| NumberOfTimeStepsInCompliancePeriod (21, 101, 201) | .. | .. | .. | .. | .. |
| NumberOfTimeStepsAfterCompliancePeriod (20, 100) | .. | .. | .. | .. | .. |
| TabularTemperatureRHFlag | .. | .. | .. | .. | .. |
| SelectRefluxModel (1, 2, 3) n.b. changes results files | .. | .. | .. | .. | .. |
| TimeOfBackfillEmplaced (100001, 100) n.b. changes results files | .. | .. | .. | .. | .. |
| FlowModelFlag | .. | .. | .. | .. | .. |
| InvertBypass | .. | .. | .. | .. | .. |
| DistanceToReceptorGroup (10, 20) n.b. use 20 for larger values of MaxTime or CompliancePeriod | .. | .. | .. | .. | .. |
| PROCESSING CONDITIONS V | | | | | |
| Name of Master | Bigbend | .. | .. | .. | .. |
| Niceness of Master Processes | 30 | .. | .. | .. | .. |
| Number of Slaves Used (# processes) | 3 | .. | .. | .. | .. |
| Names of Slaves Used: # Users | n/a | Bigbend: 6 | Hornet: 3 | Hornet: 0 | Bigbend: 10 |
| | n/a | Scratchy1: 1 | Scratchy1: 1 | --- | Scratchy1: 1 |
| | n/a | Hornet: 0 | --- | --- | Hornet: 4 |
| Niceness of Running TPA on Slaves | n/a | 25 | .. | .. | .. |
| Average Runtime of TPA on each Slave | n/a | BB: 0:50 | H: 3:00 | H: 2:20 | B: 2:10 |
| | n/a | S: 2:45 | S: 4:00 | --- | S: 4:30 |
| | n/a | H: 1:30 | --- | --- | H: 3:00 |
| Number of Jobs Per Slave | n/a | .. | H: 1 | H: 1 | BB: 48 |
| | n/a | .. | S: 1 | --- | S: 21 |
| | n/a | .. | --- | --- | H: 31 |
| Starting Time for Parjob Run (hr:min:sec) | 21:16:59 | 21:00:41 | 13:54:58 | 21:07:26 | 16:01:10 |
| Stopping Time for Parjob Run (hr:min:sec) | 21:17:22 | 21:07:17 | 13:59:24 | 21:10:12 | 17:38:11 |
| Parjob Runtime (hr:min:sec.fraction of sec) | 22.86 | 06:35.7 | 04:26.2 | 02:45.6 | 1:36:58 |
| Date of Parjob Run | 9/17/98 | 9/17/98 | 9/23/98 | 9/17/98 | 9/23/98 |
| Machine Used to Run TPA seq. | Scratchy1 | .. | .. | .. | .. |
| Niceness of Running TPA seq. | 25 | .. | .. | .. | .. |
| Number of Users | 2 | .. | .. | .. | .. |
| Starting Time for TPA seq. Run (hr:min:sec) | 6:32:25 | 0:30:09 | 0:12:36 | 0:24:28 | 17:37:28 |
| Stopping Time for TPA seq. Run (hr:min:sec) | 6:32:26 | 0:48:39 | 0:24:24 | 0:30:06 | 0:12:17 |
| TPA seq. Runtime (hr:min:sec.fraction of sec) | 1.52 | 18:30.0 | 11:47.2 | 05:38.1 | 6:34:46 |
| Date of Sequential Run | 9/27/98 | 9/26/98 | 9/26/98 | 9/26/98 | 9/26/98 |

VERIFICATION PROCESS FOR PARJOB--INPUT DECKS AND PROCESSING CONDITIONS

| FILENAMES -----> | 500real .inp | chnptime1 .inp | chnptime2 .inp | RHFlag1 .inp | FlowModel flag1.inp |
|---|--|--------------------------------|---------------------------------|---|--|
| PARAMETERS V | | | | | |
| VolcanismDisruptiveScenarioFlag (0,1) | .. | .. | .. | .. | .. |
| FaultingDisruptiveScenarioFlag (0,1) | .. | .. | .. | .. | .. |
| SeismicDisruptiveScenarioFlag (0,1) | .. | .. | .. | .. | .. |
| Subareas (one rect. zone, several subareas) | .. | .. | .. | .. | .. |
| StopAtSubarea (0 or 7,1) | | | | | |
| CheckNuclidesAndChains (0, 1) n.b. 1 should give error in tpa run on each slave | .. | .. | .. | .. | .. |
| Number of Nuclides w/i each Chain | .. | .. | .. | .. | .. |
| Number of Chains | .. | .. | .. | .. | .. |
| DirectReleaseOnlyFlag (0, 1) n.b. change only if V.D.S.Flag is 1 | .. | .. | .. | .. | .. |
| NumberOfRealizations | 500 | .. | .. | .. | .. |
| DurationOfCompliancePeriod | .. | 2.00E+04 | 2.00E+04 | .. | .. |
| MaximumTime | .. | 2.00E+04 | 1.00E+05 | .. | .. |
| NumberOfTimeStepsInCompliancePeriod (21, 101, 201) | .. | .. | 101 | .. | .. |
| NumberOfTimeStepsAfterCompliancePeriod (20, 100) | .. | .. | 100 | .. | .. |
| TabularTemperatureRHFlag | .. | .. | .. | 1 | .. |
| SelectRefluxModel (1, 2, 3) n.b. changes results files | .. | .. | .. | .. | .. |
| TimeOfBackfillEmplaced (100001, 100) n.b. changes results files | .. | 100 | 100 | .. | .. |
| FlowModelFlag | .. | .. | .. | .. | 1 |
| InvertBypass | .. | .. | .. | .. | .. |
| DistanceToReceptorGroup (10, 20) n.b. use 20 for larger values of MaxTime or CompliancePeriod | .. | 20 | 20 | .. | .. |
| PROCESSING CONDITIONS V | | | | | |
| Name of Master | Bigbend | .. | .. | .. | Scratchy1 |
| Niceness of Master Processes | 30 | .. | .. | .. | .. |
| Number of Slaves Used (# processes) | 3 | .. | .. | .. | .. |
| Names of Slaves Used: # Users | Bigbend: 11 Scratchy1: 1 Hornet: 3 | | | Bigbend: 8 Scratchy1: 1 Hornet: 0 | Bigbend: 12 Scratchy1: 2 Hornet: 2 |
| Niceness of Running TPA on Slaves | 25 | .. | .. | .. | .. |
| Average Runtime of TPA on each Slave | BB: 2:40 S: 4:00 H: 3:00 | BB: 2:10 S: 8:30 H: 3:00 | BB: 5:40 S: 12:20 H: 8:30 | BB: 2:00 S: 3:20 H: 3:00 | BB: 3:20 S: 5:00 H: 2:20 |
| Number of Jobs Per Slave | BB: 258 S: 103 H: 139 | | | BB: 4 S: 3 H: 3 | BB: 3 S: 3 H: 4 |
| Starting Time for Parjob Run (hr:min:sec) | 14:13.3 | 22:34:09 | 22:52:22 | 13:34:43 | 13:38:36 |
| Stopping Time for Parjob Run (hr:min:sec) | 23:03:36 | 22:52:12 | 23:20:27 | 13:45:38 | 13:57:11 |
| Parjob Runtime (hr:min:sec.fraction of sec) | 9:49:40 | 18:02.5 | 28:05.3 | 10:54.7 | 18:34.6 |
| Date of Parjob Run | 9/24/98 | 9/17/98 | 9/17/98 | 9/17/98 | 9/22/98 |
| Machine Used to Run TPA seq. | Scratchy1 | .. | .. | .. | .. |
| Niceness of Running TPA seq. | 25 | .. | .. | .. | .. |
| Number of Users | 2 | .. | .. | .. | .. |
| Starting Time for TPA seq. Run (hr:min:sec) | 1:07:46 | 12:53:11 | 13:56:18 | 8:05:27 | 6:33:02 |
| Stopping Time for TPA seq. Run (hr:min:sec) | 6:31:42 | 13:56:12 | 16:45:57 | 8:49:15 | 7:21:29 |
| TPA seq. Runtime (hr:min:sec.fraction of sec) | 29:23:41 | 1:03:01 | 2:49:37 | 43:48.0 | 48:27.4 |
| Date of Sequential Run | 9/26- 9/27/98 | 9/27/98 | 9/27/98 | 9/27/98 | 9/27/98 |

VERIFICATION PROCESS FOR PARJOB--INPUT DECKS AND PROCESSING CONDITIONS

| FILENAMES -----> | InvertByp Flag1.inp | Reflux Mod2.inp | Reflux Mod1.inp | StopAtSub Subarea1.inp |
|---|---|--|---|--------------------------------|
| PARAMETERS | | | | |
| V | | | | |
| VolcanismDisruptiveScenarioFlag (0,1) | .. | .. | .. | .. |
| FaultingDisruptiveScenarioFlag (0,1) | .. | .. | .. | .. |
| SeismicDisruptiveScenarioFlag (0,1) | .. | .. | .. | .. |
| Subareas (one rect. zone, several subareas) | .. | .. | .. | .. |
| StopAtSubarea (0 or 7,1) | .. | .. | .. | 1 |
| CheckNuclidesAndChains (0, 1) n.b. 1 should give error in tpa run on each slave | .. | .. | .. | .. |
| Number of Nuclides w/i each Chain | .. | .. | .. | .. |
| Number of Chains | .. | .. | .. | .. |
| DirectReleaseOnlyFlag (0, 1) n.b. change only if V.D.S.Flag is 1 | .. | .. | .. | .. |
| NumberOfRealizations | .. | .. | .. | .. |
| DurationOfCompliancePeriod | .. | .. | .. | .. |
| MaximumTime | .. | .. | .. | .. |
| NumberOfTimeStepsInCompliancePeriod (21, 101, 201) | .. | .. | .. | .. |
| NumberOfTimeStepsAfterCompliancePeriod (20, 100) | .. | .. | .. | .. |
| TabularTemperatureRHFlag | .. | .. | .. | .. |
| SelectRefluxModel (1, 2, 3) n.b. changes results files | .. | 2 | 1 | .. |
| TimeOfBackfillEmplaced (100001, 100) n.b. changes results files | .. | .. | .. | .. |
| FlowModelFlag | .. | .. | .. | .. |
| InvertBypass | 1 | .. | .. | .. |
| DistanceToReceptorGroup (10, 20) n.b. use 20 for larger values of MaxTime or CompliancePeriod | .. | .. | .. | .. |
| PROCESSING CONDITIONS | | | | |
| V | | | | |
| Name of Master | Bigbend | Scratchy1 | Bigbend | .. |
| Niceness of Master Processes | 30 | .. | .. | .. |
| Number of Slaves Used (# processes) | 3 | .. | .. | .. |
| Names of Slaves Used: # Users | Bigbend: 6 Scratchy1: 1 Hornet: 0 | Bigbend: 12 Scratchy1: 2 Hornet: 3 | Bigbend: 6 Scratchy1: 1 Hornet: 0 | .. |
| Niceness of Running TPA on Slaves | 25 | .. | .. | .. |
| Average Runtime of TPA on each Slave | BB: 1:30 S: 5:10 H: 2:40 | BB: 3:40 S: 4:30 H: 2:20 | BB: 1:30 S: 5:00 H: 2:40 | BB: 0:40 S: 2:30 H: 1:30 |
| Number of Jobs Per Slave | .. | BB: 3 S: 3 H: 4 | .. | .. |
| Starting Time for Parjob Run (hr:min:sec) | 21:53:20 | 13:57:36 | 22:04:53 | 22:27:51 |
| Stopping Time for Parjob Run (hr:min:sec) | 22:04:44 | 14:13:58 | 22:16:24 | 22:34:03 |
| Parjob Runtime (hr:min:sec.fraction of sec) | 11:23.7 | 16:21.4 | 11:30.5 | 06:12.5 |
| Date of Parjob Run | 9/17/98 | 9/22/98 | 9/17/98 | 9/17/98 |
| Machine Used to Run TPA seq. | Scratchy1 | .. | .. | .. |
| Niceness of Running TPA seq. | 25 | .. | .. | .. |
| Number of Users | 2 | .. | .. | .. |
| Starting Time for TPA seq. Run (hr:min:sec) | 7:21:35 | 9:31:26 | 8:49:21 | 10:15:14 |
| Stopping Time for TPA seq. Run (hr:min:sec) | 8:05:21 | 10:15:08 | 9:31:20 | 10:30:42 |
| TPA seq. Runtime (hr:min:sec.fraction of sec) | 43:45.7 | 43:42.0 | 41:58.9 | 15:27.6 |
| Date of Sequential Run | 9/27/98 | 9/27/98 | 9/27/98 | 9/27/98 |

PLEASE NOTE:

| | |
|-----|---|
| .. | indicates that the input in this cell is identical to the corresponding cell of base.inp |
| n/a | indicates that input file produced a failure in running TPA on each slave; therefore, this cell is not applicable |
| --- | indicates that there were a greater number of slave processes than there were realizations to run |

FOR ALL INPUT DECKS: OuterWPThickness = 0.001

InnerWPThickness[m] = 0.002

FOR chngtime2.inp: RatioOfLastToFirstTimeStepAfterCompliancePeriod = 100.00

ALL INPUT CONDITIONS were provided by Ron Janetzke of the Nuclear Research Center, SwRI, and can be found in the file *postscript*

1.7 Troubleshooting

1.7.1 Reading Slave Results

PVM writes the parjob-related standard output from all Slaves to a file named “/tmp/pvml.uid”, on the local temp directory of the host on which parjob.exe was invoked, where uid is an integer unique to each user.

All TPA output on the Slaves is written to a file “tpa.mode2” contained in each working directory

1.7.2 Error running PVM

Two simple approaches. Try halting PVM then restarting. Type “pvm” to enter the PVM console. Type “halt” to exit PVM.

A second approach is to delete the pvm* files located on the temporary disk of each host. Try deleting the pvml* and pvmd* files on the /tmp directory of the host. This is accomplished easily using a remote shell command. For example, to delete file /tmp/pvml.203 on host Saturn, use the command

```
remsh saturn 'rm /tmp/pvm*'
```

1.7.3 Can't add host

Try deleting the pvml* and pvmd* files on the /tmp directory of the host. This is accomplished easily using a remote shell command. For example, to delete file /tmp/pvml.203 on host Saturn, use the command

```
remsh saturn 'rm /tmp/pvm*'
```

1.7.4 TPA Aborts on Slave

It may happen the TPA will abort on the Slave. This may happen when TPA is being run in conjunction with other large programs on the same machine. There is no known general solution to this. The file tpa.mode2 can be queried to determine the cause of failure. Often, TPA will run correctly on the Slave at a later time.

1.8 Acknowledgements

This manual was written by Harry Millwater and Nathan Kelley- Probabilistic Mechanics and Reliability Section, Southwest Research Institute. Significant contributions were made by Clay Cowan - coop student from The University of Kentucky, and Ron Janetzke - Center for Nuclear Waste Regulatory Analysis, Southwest Research Institute.

1.9 References

Millwater, H., Wu, J., and Thacker, B., "Efficient Uncertainty and Sensitivity Analysis Methods for Performance Assessment of Complex Systems", SwRI Project 06-9994, September 1998
Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Mancheck, R., Sunderam, V., "PVM: Parallel Virtual Machine, A User's Guide and Tutorial for Networked Parallel Computing," MIT Press, Cambridge, Massachusetts, London, England, 1994

TPA-PVM SOFTWARE

Feasibility Plan

May 7, 1998

Objective

The TPA code performs a nuclear waste assessment considering uncertainties in a number of inputs. A Latin Hypercube Sampling methodology is used to vary the inputs and rerun the analysis. As such, a number of analyses are required, each independent with different inputs. PVM (Parallel Virtual Machine) is a free public-domain software library that can be used to network UNIX workstations into a pseudo parallel processing machine and perform the analyses in parallel, thus reducing computational time. This project would integrate features of PVM with TPA version 3.2 in order to compute the results for a series of input vectors in a more computationally efficient manner by performing the computations in parallel. The advantage of this approach is that it utilizes existing hardware, UNIX workstations, and existing software. In addition, much of the coordinating FORTRAN code has already been developed at SwRI under Internal Research funding.

Capability

The following software modifications would be made:

- modify TPA executive to
 - call LHS module
 - call PVM Master module
 - call TPA postprocessing module
- use PVM master-slave module from focused IR to
 - spawn slaves on workstation network
 - send vectors to each slave
- slave calls TPA for single vector analysis
- slave collects single vector results files and passes back to master
- master receives single vector results
- master concatenates all relevant results into a single file that is identical to a serial (non-PVM) analysis
- TPA executive is run to post-process results

Estimated Level of Effort/Estimated Duration

| | |
|--|-----------------|
| Modify TPA executive: - | 5 weeks |
| Integrate PVM master-slave module - | 2 weeks |
| Modify slave to call TPA for single vector analysis - | 2 weeks |
| Modify slave to collect TPA results and send back to master - | 1 week |
| Modify master to concatenate all results into single result file - | 3 weeks |
| Documentation | 4 weeks |
| Verification | 3 weeks |
| Installation/Travel | 1 week |
| Total | 21 weeks |

Principal Personnel

Personnel required for this effort include:

- Programmer-
 - Institute staff (33%)
 - Summer student (100%)

- TPA user for consulting/testing Institute staff (10%)

TPA 3.2 - PVM Project

Coding Modifications

1. Modify TPA executive
 - a. Call LHS module
 - b. Call PVM Master module
 - c. Call TPA postprocessing module
2. Use PVM master-slave module from focused IR
 - a. Spawns slaves on workstation network
 - b. 1 vector sent to each slave
3. Slave calls TPA for single vector analysis
4. Slave collects single vector results files and passes back to master
5. Master receives single vector results
6. Master concatenates all relevant results into a single file that is identical to a serial (non-PVM) analysis
7. Run TPA executive to post process results

Documentation

User's manual (not extensive, ~5 pages)

Verification

2 representative TPA analyses

Installation

1 trip to NRC to install and verify

Costs

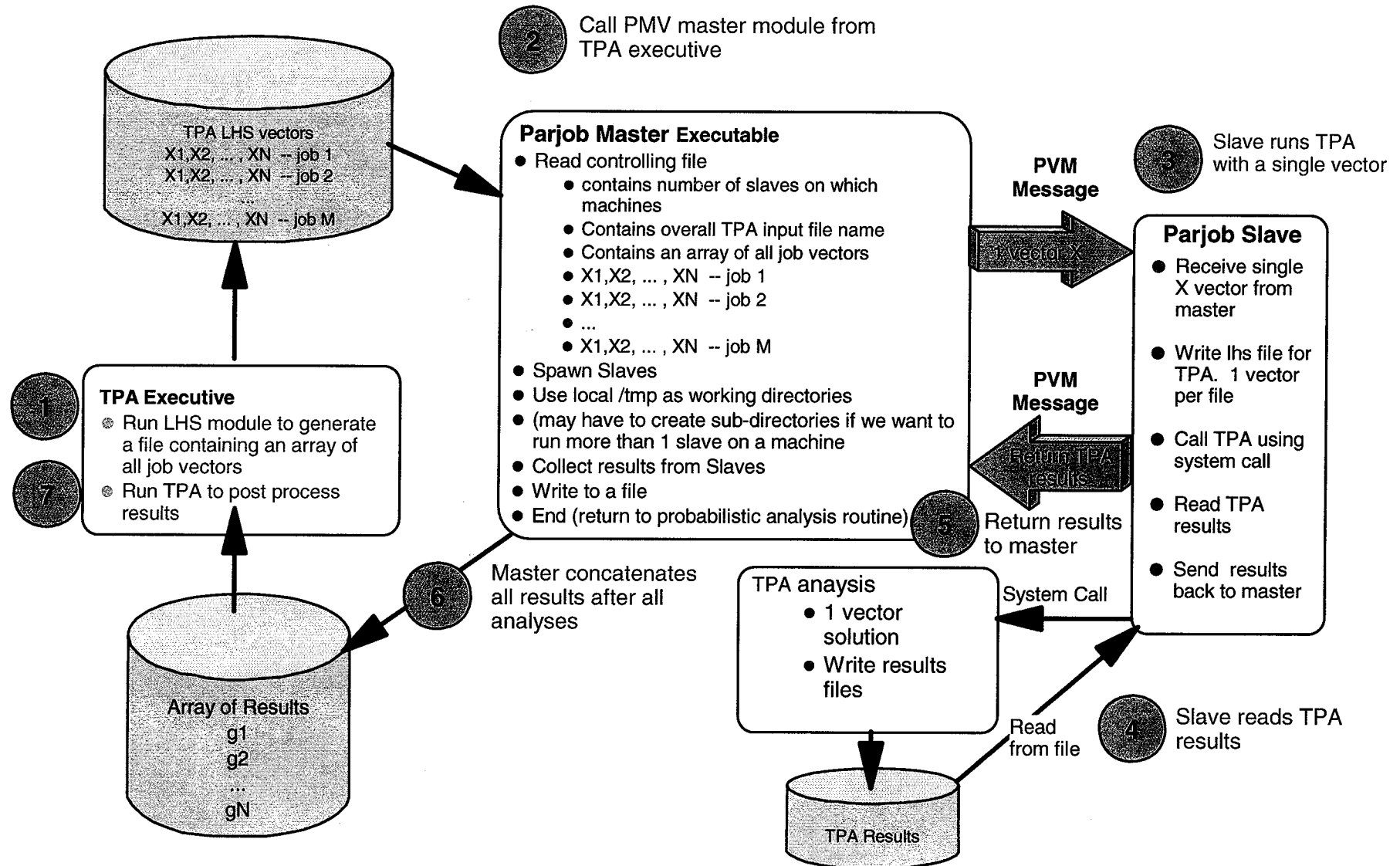
8 man-weeks PL1
5 man-weeks PL2
1 Trip to NRC

Schedule

Completion by
September 31, 1998

57/63

TPA 3.2 PVM Modification



60/63


```
a parjob2/ 0 tape blocks
a parjob2/Makefile 5 tape blocks
a parjob2/copyinp 1 tape blocks
a parjob2/get_result.f 31 tape blocks
a parjob2/parjob_loc.f 6 tape blocks
a parjob2/run_job.f 23 tape blocks
a parjob2/tpa_loc.f 10 tape blocks
a parjob2/write_result.f 14 tape blocks
a parjob2/parjob.a 688 tape blocks
a parjob2/fpvm3.h 18 tape blocks
a parjob2/file_names.i 6 tape blocks
a tpapvm/ 0 tape blocks
a tpapvm/CLEANUP 3 tape blocks
a tpapvm/array.f 56 tape blocks
a tpapvm/ashplumo.f 40 tape blocks
a tpapvm/ashrmovo.f 47 tape blocks
a tpapvm/condxyzt.f 20 tape blocks
a tpapvm/dcags.f 44 tape blocks
a tpapvm/dcagw.f 87 tape blocks
a tpapvm/ebsfail.f 68 tape blocks
a tpapvm/ebsrel.f 122 tape blocks
a tpapvm/exec.f 606 tape blocks
a tpapvm/execa.i 4 tape blocks
a tpapvm/execb.i 1 tape blocks
a tpapvm/faulto.f 22 tape blocks
a tpapvm/fileunit.f 12 tape blocks
a tpapvm/findelev.f 12 tape blocks
a tpapvm/invent.f 108 tape blocks
a tpapvm/Makefile 2 tape blocks
a tpapvm/max500yr.i 1 tape blocks
a tpapvm/maxchain.i 1 tape blocks
a tpapvm/maxnnucl.i 1 tape blocks
a tpapvm/maxnsuba.i 1 tape blocks
a tpapvm/maxntime.i 1 tape blocks
a tpapvm/mv.f 23 tape blocks
a tpapvm/nfenv.f 146 tape blocks
a tpapvm/reflux2.i 1 tape blocks
a tpapvm/nintv.i 1 tape blocks
a tpapvm/notice.i 3 tape blocks
a tpapvm/numrecip.f 10 tape blocks
a tpapvm/path.i 1 tape blocks
a tpapvm/peakfinder.f 13 tape blocks
a tpapvm/ran.f 86 tape blocks
a tpapvm/reader.f 153 tape blocks
a tpapvm/reader.i 1 tape blocks
a tpapvm/sampler.f 166 tape blocks
a tpapvm/seismo.f 109 tape blocks
a tpapvm/setfiles.i 4 tape blocks
a tpapvm/stop.i 1 tape blocks
a tpapvm/subarea.f 69 tape blocks
a tpapvm/szft.f 135 tape blocks
a tpapvm/szft.i 1 tape blocks
a tpapvm/uzflow.f 108 tape blocks
a tpapvm/uzft.f 219 tape blocks
a tpapvm/volcano.f 21 tape blocks
a tpapvm/ful.i 1 tape blocks
a tpapvm/fu2.i 1 tape blocks
a tpapvm/inventa.i 1 tape blocks
a tpapvm/inventb.i 1 tape blocks
a tpapvm/inventc.i 1 tape blocks
a tpapvm/inventd.i 1 tape blocks
a tpapvm/invente.i 1 tape blocks
a tpapvm/inventf.i 1 tape blocks
a tpapvm/inventg.i 1 tape blocks
a tpapvm/inventh.i 1 tape blocks
```

```
a tpapvm/inventi.i 1 tape blocks
a tpapvm/inventj.i 1 tape blocks
a tpapvm/inventk.i 1 tape blocks
a tpapvm/inventl.i 1 tape blocks
a tpapvm/inventm.i 1 tape blocks
a tpapvm/inventn.i 1 tape blocks
a tpapvm/mva.i 1 tape blocks
a tpapvm/mvb.i 1 tape blocks
a tpapvm/mvc.i 1 tape blocks
a tpapvm/mvd.i 1 tape blocks
a tpapvm/mve.i 1 tape blocks
a tpapvm/mvf.i 1 tape blocks
a tpapvm/sampler0.i 1 tape blocks
a tpapvm/sampler1.i 1 tape blocks
a tpapvm/sampler2.i 1 tape blocks
a tpapvm/sampler3.i 1 tape blocks
a tpapvm/sampler.a.i 1 tape blocks
a tpapvm/samplerb.i 1 tape blocks
a tpapvm/samplerc.i 1 tape blocks
a tpapvm/samplerd.i 1 tape blocks
a tpapvm/sampler.e.i 1 tape blocks
a tpapvm/samplerf.i 1 tape blocks
a tpapvm/samplerg.i 1 tape blocks
a tpapvm/samplerh.i 1 tape blocks
a tpapvm/sampleri.i 1 tape blocks
a tpapvm/samplerj.i 1 tape blocks
a tpapvm/codes/ 0 tape blocks
a tpapvm/codes/Makefile 1 tape blocks
a tpapvm/codes/README 1 tape blocks
a tpapvm/codes/SIZES.INC 5 tape blocks
a tpapvm/codes/SIZES2.INC 1 tape blocks
a tpapvm/codes/ashplume.f 185 tape blocks
a tpapvm/codes/fault.f 157 tape blocks
a tpapvm/codes/nefms.f 591 tape blocks
a tpapvm/codes/reaset.f 337 tape blocks
a tpapvm/codes/snllhs.f 335 tape blocks
a tpapvm/codes/ebsfilt.f 19 tape blocks
a tpapvm/samplerk.i 1 tape blocks
a tpapvm/samplerl.i 1 tape blocks
a tpapvm/samplerm.i 1 tape blocks
a tpapvm/samlern.i 1 tape blocks
a tpapvm/sampler.o.i 1 tape blocks
a tpapvm/samplerp.i 1 tape blocks
a tpapvm/samplerq.i 1 tape blocks
a tpapvm/samlerr.i 1 tape blocks
a tpapvm/samplers.i 1 tape blocks
a tpapvm/samlert.i 1 tape blocks
a tpapvm/sampleru.i 1 tape blocks
a tpapvm/samlerv.i 1 tape blocks
a tpapvm/samplerw.i 1 tape blocks
a tpapvm/samlerox.i 1 tape blocks
a tpapvm/samlery.i 1 tape blocks
a tpapvm/samplerz.i 1 tape blocks
a tpapvm/subareaa.i 1 tape blocks
a tpapvm/subareab.i 1 tape blocks
a tpapvm/subareac.i 1 tape blocks
a tpapvm/subaread.i 1 tape blocks
a tpapvm/subareae.i 1 tape blocks
a tpapvm/uz_climi.i 1 tape blocks
a tpapvm/uz_climr.i 2 tape blocks
a tpapvm/uz_climz.i 1 tape blocks
a tpapvm/uz_flowi.i 1 tape blocks
a tpapvm/uz_flowr.i 2 tape blocks
a tpapvm/uz_flowz.i 1 tape blocks
a tpapvm/uz_parms.i 6 tape blocks
```

```
a tpapvm/data/ 0 tape blocks
a tpapvm/data/multiflo.dat 2491 tape blocks
a tpapvm/data/strmtube.dat 9 tape blocks
a tpapvm/data/climato1.dat 1661 tape blocks
a tpapvm/data/climato2.dat 5 tape blocks
a tpapvm/data/dilution.dat 4 tape blocks
a tpapvm/data/ebsfail.def 9 tape blocks
a tpapvm/data/ebsrel.def 9 tape blocks
a tpapvm/data/rectedge.dat 5 tape blocks
a tpapvm/data/soildem.dat 965 tape blocks
a tpapvm/data/elevdem.dat 589 tape blocks
a tpapvm/data/gs_cb_ad.dat 6 tape blocks
a tpapvm/data/gs_cb_ci.dat 5 tape blocks
a tpapvm/data/gs_pb_ad.dat 6 tape blocks
a tpapvm/data/gs_pb_ci.dat 5 tape blocks
a tpapvm/data/gw_cb_ad.dat 7 tape blocks
a tpapvm/data/gw_cb_ci.dat 3 tape blocks
a tpapvm/data/gw_pb_ad.dat 7 tape blocks
a tpapvm/data/gw_pb_ci.dat 3 tape blocks
a tpapvm/data/tefktd.inp 1473 tape blocks
a tpapvm/data/tpanames.dbs 182 tape blocks
a tpapvm/data/ebsfilt.def 2 tape blocks
a tpapvm/data/dry_thick.dat 1 tape blocks
a tpapvm/ccdf/ 0 tape blocks
a tpapvm/ccdf/tccdf.f 45 tape blocks
a tpapvm/ccdf/tccdf.i 1 tape blocks
a tpapvm/ccdf/tccdf.inp 2 tape blocks
a tpapvm/ccdf/Makefile 1 tape blocks
```