

SCIENTIFIC NOTEBOOK

170-18E

Printed:

R. Rogers

SCIENTIFIC NOTEBOOK No. 170-18e

SCIENTIFIC NOTEBOOK

by

Robert Rogers

Southwest Research Institute
Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

Printed:

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

INITIAL ENTRIES

Scientific Notebook: #170-18e

Issued to: S. Mohanty

Issue Date: June 30, 2002

Account Number: 20-1402-762

Title: Porting TPA Software to GNU Fortran compiler under Red Hat Linux

Participants: Robert Rogers, Ron Janetzke

July 8, 2002 - Modifications to TPA Software for Linux Conversion

Using the latest Unix version of the TPA software (tpa42f), I started making modifications to the software based on errors from the GNU Fortran compiler (g77 version 2.96) running under Red Hat Linux 7.1, on a Compaq Deskpro (pentium II, 400 MHz) computer.

July 11, 2002 - Debugging continues

Received the following software changes in an e-mail from Ron Janetzke (changes that Mike Muller had made to the Unix version of TPA 4.1 to run it under SuSE Linux and Windows NT using the Lahey-Fujitsu Fortran compiler version LF95):

Sep. 18, 2001. Michael Muller, SwRI.
Changes made to code to work under lf95 on NT-PC and lf95 on linux-PC.

ebsrel.f 7 changes.

738

```
> cc 4/20/01 mam modified write from float to int because
> cc                    lf95 reads ints as ints.
```

Printed:

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

```
741
c      write( aline(1:24),fmt='(2(1pe12.5))' )
      write( aline(1:24),fmt='(1pe12.5,i7,5x))' )
&      releasewpfailedtime(2,1), int(releasewpfailedtime(1,1)) !init
746
c      write( aline(1:24),fmt='(2(1pe12.5))' )
      write( aline(1:24),fmt='(1pe12.5,i7,5x))' )
&      releasewpfailedtime(2,2), int(releasewpfailedtime(1,2)) !fault
751
c      write( aline(1:24),fmt='(2(1pe12.5))' )
      write( aline(1:24),fmt='(1pe12.5,i7,5x))' )
&      releasewpfailedtime(2,3), int(releasewpfailedtime(1,3)) !volcano
756
c      write( aline(1:24),fmt='(2(1pe12.5))' )
      write( aline(1:24),fmt='(1pe12.5,i7,5x))' )
&      releasewpfailedtime(2,4), int(releasewpfailedtime(1,4)) !seismo1
761
c      write( aline(1:24),fmt='(2(1pe12.5))' )
      write( aline(1:24),fmt='(1pe12.5,i7,5x))' )
&      releasewpfailedtime(2,5), int(releasewpfailedtime(1,5)) !seismo2
766
c      write( aline(1:24),fmt='(2(1pe12.5))' )
      write( aline(1:24),fmt='(1pe12.5,i7,5x))' )
&      releasewpfailedtime(2,6), int(releasewpfailedtime(1,6)) !seismo3
771
c      write( aline(1:24),fmt='(2(1pe12.5))' )
      write( aline(1:24),fmt='(1pe12.5,i7,5x))' )
&      releasewpfailedtime(2,7), int(releasewpfailedtime(1,7)) !seismo4
```

Deleted these two lines from files:

```
integer sh
external sh
```

```
ashplumo.f
dcags.f
nfenf.f
sampler.f
seismo.f
szft.f
uzft.f
```

Deleted these two lines from files:

```
integer system
external system
```

```
sampler.f
uzft.f
```

zportlnx.f (changes for lf95 and linux).

74

DOUBLE PRECISION DERF

Printed:

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

```
C      EXTERNAL DERF
77
c      Call Sun's d_erf
c      ZPORTDERF=D_ERF(X)
c      Call Lahey/Fujitsu's derf
c      ZPORTDERF=DERF(X)

169
C      INTEGER ieee_flags
C      EXTERNAL ieee_flags
173
C      ZPORTIEEE_FLAGS=ieee_flags(myaction,mode,inin,outout)
C      Return 1 = mode not supported.
C      ZPORTIEEE_FLAGS=1

224
C      INTEGER ieee_handler
C      EXTERNAL ieee_handler

C      ZPORTIEEE_HANDLER=ieee_handler(myaction,exception,HANDLER)
C      Return 1 = exception not supported.
227
C      ZPORTIEEE_HANDLER=ieee_handler(myaction,exception,HANDLER)
C      ZPORTIEEE_HANDLER=1

zportpc (removed 1 c in comment, does not have to be done.)
92
c      WRITE(*,*) 'IN ZPORTSH, COMMAND:',COMMAND
94
c      WRITE(*,*) '                DOS EQ:',NEWCMND
460
c      CHARACTER*32 GETENV
522
c      CALL GETENV(ENVIRON(1:LEN_TRIM(ENVIRON)),TOKENVAL)
c      TOKENVAL=GETENV(ENVIRON)
620
c      CHARACTER*32 GETENV
677
c      CALL GETENV(ENVIRON(1:LEN_TRIM(ENVIRON)),TOKENVAL)
c      TOKENVAL=GETENV(ENVIRON)

codes/gentpa/makda2.f

16
c      INTEGER ihms(3)
19
C      M.Muller: Replace with equivalent:
C      CALL TIME (CLOCK)

c      CALL ITIME(ihms)
c      WRITE(CLOCK,'(I2.2,":",I2.2,":",I2.2)') ihms(1),ihms(2),ihms(3)
```

July 19, 2002 - Debugging continues

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

Had problems with variable values changing between subroutine calls. Researched the compiler flags used by M. Muller under lf95. Only a few of the compiler options are available under the GNU compiler. Set the “-fno-automatic” flag to save variable values between calls. There is a problem with a COMMON block in ebsfail.c, where the variable is mistyped: ii_append_upper should be iiappend_upper. This error is not noticed with the “-fno-automatic” flag set.

July 24, 2002 - Debugging continues

After several tweaks to the code and tpa.inp, the software now runs up to subarea 10 of 10 on realization 1, but it fails on an “open()” command in dcagw.f, subroutine gentpa. The fix was to change “igetunitnumber()” in fileunit.f, so that it wraps around after reaching the unit number 99. This causes other problems later on. To fix this problem would be a significant rewrite of the code. Ron J makes the decision to halt this effort and switch over to the lf95 compiler on masaya@geophysics.swri.edu.

July 30, 2002 - Linux modifications completed

Starting over again with the Unix version of the TPA software (tpa42f), Muller’s changes listed above were incorporated, along with his version of the Makefiles. His makefiles needed only a minor update to include dsfail.o, dsfalt.o, integrt.o, linintrp.o, and srchpos.o. With these changes the compiler and runs successfully.

The timing results of a 100,000-year (1 realization, 10 subareas, no disruptive events) run were:

real 6m22.338s
user 2m56.980s
sys 0m9.730s

The timing results of a 10,000-year (1 realization, 10 subareas, no disruptive events) run were:

real 2m56.249s
user 1m18.650s

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

sys 0m6.020s

Title: Task ID D4, Use GENTPA for DCAGS dose conversion factors (DCFs).

Participants: Robert Rogers, Mike Smith

August 2, 2002 -

In telephone conference with Mike Smith and Pat LaPlante, we discussed the changes needed for dcags.f to generate different DCFs for each realization. The "gentpa()" and "gentodcf()" subroutines in dcagw.f needed to be duplicated in dcags.f. Mike and Pat were going to generate versions of the following dcagw input files for dcags: ggenii.def (new file: ggeniis.def), gdefault.def (new file: gdefaults.def), and gftrans.def (new file: gftranss.def). They also decided to only provide the option to append data to the file dcf.cum and not to genii.cum or genv.cum.

They had a new formula for computing the external exposure time that need to be incorporated into dcags:

$(1+2) + f*(3+4) = \text{external exposure time}$, where 1-4 refer to the following:

- 1) 'OccupancyFractionOutsideHeavyDisturbance[-]' = 0.01 [0,1]
- 2) 'OccupancyFractionOutsideLightDisturbance[-]' = 0.20 [0,1]
- 3) 'OccupancyFractionInsideHeavyDisturbance[-]' = 0.24 [0,1]
- 4) 'OccupancyFractionInsideLightDisturbance[-]' = 0.55 [0,1]
- 5) 'OccupancyFractionOffsite[-]' = 0 [0,1]

and f is the shielding factor for building materials (make f=0.5 for now).

The sum of the occupancy fractions would be required to equal one. Note: [x,y] denotes the minimum, x, and maximum, y, values allowed for each parameter.

August 5, 2002 -

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

Modified dcags.f by adding gentpa(), gentodcf(), and copylines() from dcagw.f. Changed the following variable and file names in these functions from use in dcags.f:

```
ggenii.def -> ggeniis.def
gdefault.def -> gdefaults.def
gftrans.def -> gftranss.def
igeniidef -> igeniisdef
idefaultdef -> idefaultsdef
iftransdef -> iftranssdef
```

```
gw_cb_ad.dat -> gs_cb_ad.dat
gw_cb_ci.dat -> gs_cb_ci.dat
gw_pb_ad.dat -> gs_pb_ad.dat
gw_pb_ci.dat -> gs_pb_ci.dat
igwaddat -> igsaddat
igwcidat -> igscidat
```

August 6, 2002 –

During a compilation of dcags.f on masaya, the following multiple definitions were found:

```
gentodcf() –changed to—> gentodcfgs()
copylines() –changed to—> copylinesgs()
gentpa() –changed to—> gentpags()
```

In exec.f, added ir (irl in dcags()) to the call to dcags() to track the realization number, as done in dcagw().

Removed the copy commands for the old “gs” DCF files; they copied the “gs” DCF files from the /data directory to the current directory. These old files will no longer be needed; their replacements are generated by the new subroutines we have just put into dcags().

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

Commented out where dcags() appends data to ggenii.cum and genv.cum. These options may be used later but are not currently desired.

August 7, 2002 –

Out of a discussion with Mike Smith on D4:

- 1) Change menu option 20 to also indicate GS;
- 2) Output a dcf_gw.cum and a dcf_gs.cum file instead of the single dcf.cum;
- 3) Read shielding factor from tpa.inp and use in calibration for new SoilContaminationExposureTime value in ggeniis.inp.

August 8, 2002 –

Added IndoorShieldingFactor to tpa.inp and indoorshielding variable to dcags(). The following three lines were added to tpa.inp after RateOfReductionOfMassLoadingFactor[1/yr] in the DCAGS section:

```
constant
IndoorShieldingFactor
0.7
```

No change is required to ispquery() or related subroutines or data structures to add this new constant to tpa.inp. ispquery() uses the following array description in samplerf.i:

```
character*(mnznam) description
common/pdfdb5/description(maxpdf)
```

The following are the current values in the listed files; these values will accommodate the addition of IndoorShieldingFactor to tpa.inp:

| | |
|---------------------------|-----------------------|
| parameter (mnznam = 60) | defined in samplerw.i |
| parameter (maxpdf = 2000) | defined in samplerv.i |

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

“ing milk” was added to gs_cb_ad.dat and gs_pb_ad.dat, as well as to the dcf(k) formula in dcags().

August 9, 2002 –

With the additional files being opened by dcags(), the maxunit constant defined in fu2.i and used by fileunit.f had to be increased – value was set to 200.

For testing purposes: There is an if branch in dcags() that has to be forced to true in order to switch between reading gs_cb_ad.dat and gs_pb_ad.dat:

if(dMAP(it) .lt. dKGboundary)

replaced with:

if(0 .lt. 1) or similar

This check is comparing if the mean annual precipitation (MAP) goes below some mean annual temperature:

dKGboundary = 0.22 * (dMAT(it) – 32.0)

August 19, 2002 –

Sent software for D4/381 to Mike Smith for review

August 22, 2002 –

The inhalation values in the new gs_cb_ad.dat seem to be off by the power of +14. Mike Smith will check with Pat on the adjustment factor (ADJUNT) we are using in dcags.f and check on the values in genv.out from dcags.f.

Title: Task ID EX4, Create a new output file that contains the pathway specific doses.

Participants: Robert Rogers, Mike Smith

August 19, 2002 –

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

For EX4, they want an output file generated by dcagw() which tracks the doses by nuclide. Right now the code has the following arrays:

 dcfde(k), dcfinh(k), dcfinga(k), dcfingp(k), dcfdw(k), dcfmlk(k)
and
 ciperyrallsafromsz(j,k)

August 22, 2002 –

For EX4, the new preliminary design will tentatively loop at two outputs (for dcagw()):

1) for each of the six pathways (de, inh, inga, ingp, dw, mlk), output the value below for each nuclide (i.e., k) and each time step:

 remperryrpernuclgw per pathway(j,k) =
 dcf____(ik) * ciperyrallsafromsz(j,k) / dilutionvolume *
 fractionmasscaptured

2) for each of the six pathways
 for each of the nuclides
 output the peak concentration for the entire time period

 append the output to dose.cum

The number of time steps is defined by NumberOfTimeStepsInCompliancePeriod in tpa.inp, currently set to 100.

The number of time steps is also defined as the constant above +
NumberOfTimeStepsAfterCompliancePeriod.

August 26, 2002 –

Change in preliminary design: Do not capture the peak values, as previously discussed, and do not create the dose.cum file. Instead create two dose output files for EX4: rgwsap.tpa and rgwnap.tpa, to be produced by exec() like the files rgwsa.tpa and rgwna.tpa.

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

Will need to pass a new array from exec() to subroutine call to dcagw().

In exec(), use the following new arrays:

```
remperyrpwnr( maxntime, maxnnucl, maxnpathgw )  
remperyrpwnap( maxntime, maxnnucl, maxnpathgw )  
remperyrpwsap( maxntime, maxnpathgw )
```

where:

maxntime is defined in maxntime.i as 401

maxnnucl is defined in maxnnucl.i as 43

and the new parameter maxnpathgw is defined as 6

Add remperyrpwnr to call to dcagw() in exec().

Add remperyrpwnuclperpathgw to dcagw() definition.

In exec(), add the handling of these new arrays in similar locations as the following:

```
rgwna.tpa<- remperyrpwna(maxntime,maxnnucl)  
rgwsa.tpa<- remperyrpwsa(maxntime)
```

```
call zero(maxntime*maxnnucl, remperyrpwna(1,1))  
call zero(maxntime, remperyrpwsa(1))
```

```
writehead2(irgwna, ..., "rgwna.tpa")  
writehead(irgwsa, ..., "rgwsa.tpa")
```

```
call addto(..., remperyrpwnr(1,1), remperyrpwna(1,1))
```

```
part of write to "check.pnt"  
part of read from "check.pnt"
```

```
call scale(...)  
writedata(irgwna, ...)
```

Printed:

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

Out of a conversation with Ron Janetzke and Mike Smith: Create six new files, one per pathway, instead of the one file rgwnap.tpa. Use the pathway categories in the file names: ext, inh, ani, pla, dw, milk. The pathnames from genv.out are separated out into these 6 categories.

August 28, 2002 –

Imported the new data files (rgwnap*.tpa and rgwsap.tpa) and the old data files (rgwna.tpa and rgwsa.tpa) into MicroSoft Excel to verify that data summed up correctly to match between the new and the old. A small typographical error was found.

Title: Task ID IA3, Add ash redistribution model.

Participants: Robert Rogers, Mike Smith

September 3-6, 2002 –

Discussed with Mike Smith the formulas generated by James Weldy and how they would be applied to the software.

Title: Task ID EX1, Add limited flexibility to exec for variations to the DOE current thermal loading strategy.

Participants: Robert Rogers, Ron Janetzke

August 30, 2002 –

The drifts() calculator in reader.f determines the endpoints of the drifts, the number of WP in each drift, and the fill order based on emplacement blocks. All of this info is written into drifts.dat.

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

drifts() determines the number of WP using the following:

$$\text{amtu} = \text{amtu} - \text{numWP} * \text{amtuperwp}$$

$$\text{numWP} = (\text{amtuperrep} - \text{amtu}) / \text{amtuperwp}$$

where

amtuperwp is 'WastePackagePayload[MTU]'

amtuperrep is 'TotalWasteEmplacedInRepository[MTU]'

also

$$\text{numWP} = \text{driftlength} / \text{Wpspacing}$$

Need to calculate the (x,y) endpoints for both ends of WP.

September 12, 2002 –

From meeting with Ron on creating a new "WP endpoints" version of drifts.dat:

1) increase the max. number of drifts to 10,000 (in driftsa.i);

2) readjust the computation logic I was using in drifts();

3) change nfenv() to use a second read inside the

do n=1,tbump

loop to go back through drifts.dat one line at a time and calculate a new tbump.

Need to cut out the large arrays that are sized based on number of drifts

'NumberOfDriftsInRepository[]' which is set by drifts().

September 13, 2002 –

Making change drifts() to output new formatted drifts.dat where each WP is considered a drift. drifts() sets kntdrifts to keep track of the number of drifts (which is the number of WPs in this case) and sets 'NumberOfDriftsInRepository[]', which is only used by nfenv() to process the heat.

With the current settings:

- drifts() is calculating the number of WP as 8920

- fillsubareas() is calculating the total number of WP as 8872 (in 10 subareas), a loss of 48 WP

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

Need to check if this difference is acceptable:

 gsanwp() in subarea.f returns nwpsa(isa), or nwpsastore(isa), for subare isa.

 gsanwpglass() uses nwpsastore(isa) to calculate amtupersa, the MTU for the subarea, which is used to calculate the returned variables nglasspersa and nwppersa.

 reader() uses the amtusa passed back through drifts() to calculate and set 'ArealMassLoading[MTU/acre]'.

 nfenv() uses 'DriftArealMassLoading[MTU/acre]' instead of 'ArealMassLoading[MTU/acre]' due a recent change in the code, so there is a reason to track the two different numbers. Therefore, the difference in WPs above is expected.

maxnumdrifts in driftsa.i is only used by fillsubareas() in reader.f and by nfenv(). Increasing this value to 10,000 only effects the sizes of following arrays in fillsubareas():

 drxy1(2,maxnumdrifts,maxnumempblks)

 drxy2(2,maxnumdrifts,maxnumempblks)

 numWP(maxnumdrifts,maxnumempblks)

and in nfenv():

 drxy1(2,maxnumdrifts,maxnumempblks)

 drxy2(2,maxnumdrifts,maxnumempblks)

 drxy1sa(2,maxnumdrifts*maxnumempblks)

 drxy2sa(2,maxnumdrifts*maxnumempblks)

 numWP(maxnumdrifts,maxnumempblks)

In nfenv(), the arrays drxy1 and drxy2 can be deleted, and drxy1sa and drxy2sa can be used in their place, which will save considerable memory space. drxy1sa and drxy2sa can be reduced in size to the following:

 drxy1sa(2,maxnumdrifts)

 drxy2sa(2,maxnumdrifts)

The array numWP is not currently used and could be deleted as well.

September 17, 2002 –

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

A size limit on the arrays xc and yc of 300 was causing overwrite problems since these arrays are being written to for the index range of 1 to 8920 (the total number of drifts). Since these variables do not need to be arrays, I changed them to regular variables to save memory and to eliminate the array bounds problem.

The arrays aLs and aBs are not used, so they were deleted.

September 19, 2002 –

An addition change for EX1 (SCR386): change the value of ``DriftArealMassLoading[MTU/acre]``, which is set by `drifts()`, to reflect the percentage of waste packages in glass form. This change is currently only for the thermal analysis performed in `nfenv()`, where ``DriftArealMassLoading[MTU/acre]`` is used.

The specific change consists of a conditional assignment to the variable `amtuperwp`. If the flag ``OneTemperatureCellPerWP...`` is zero, `amtuperwp` will equal the value of ``WastePackagePayload[MTU]``; otherwise, it will equal the following:

$$\text{amtuperwp} = (1 - \text{fraction_of_glass_in_repository}) * \text{MTU_for_SF} + \text{fraction_of_glass_in_repository} * \text{MTU_for_glass}$$

This value for `amtuperwp` is also passed back to `reader()` as part of the computed values in the array `amtusa` and submitted to the subarea database in the call to `ssadba()`.

Future changes may be required in other areas of the code where the value of ``WastePackagePayload[MTU]`` is used, to reflect the effect of glass WPs in other areas of analysis.

A comment was added to `tpa.inp` to indicate that the value for `fraction_of_glass_in_repository` should not be set above 0.24, otherwise not enough WPs will be put in the repository to meet the MTU requirement. An error message in `reader()` will be displayed if the buried MTU for the repository is less than 60,000 MTU.

Printed: September 20, 2002

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

ssadba().

Future changes may be required in other areas of the code where the value of 'WastePackagePayload[MTU]' is used, to reflect the effect of glass WPs in other areas of analysis.

A comment was added to tpa.inp to indicate that the value for fraction_of_glass_in_repository should not be set above 0.24, otherwise not enough WPs will be put in the repository to meet the MTU requirement. An error message in reader() will be displayed if the buried MTU for the repository is less than 60,000 MTU.

Entries into Scientific Notebook #170-18e for pages 1-17 have been made by Robert Rogers 9/20/02.

No original text entered into this Scientific Notebook has been removed.

 9/20/02.

I have reviewed this scientific notebook and find it in compliance with QAP-001. There is sufficient information regarding methods used for conducting tests, acquiring and analyzing data so that another qualified individual could repeat the activity.

Printed:

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

Title: TPA Code Changes and Testing

Participant: Robert Rogers

October 7, 2002 –

Submitted Linux versions of makefiles to be added to the repository. Also included zportlnx.f and makda2lnx.f.

October 8, 2002 –

SCR-399 involves a time-dependent dilution volume change to dcagw.f and exec.f; and was put into version 4.2w.

Changes to exec.f (line numbers refer to version 5.0beta):

| | | |
|------|--------------------------------------|---------------------|
| 424 | double precision dilvolume(maxntime) | changed to an array |
| 4947 | used in formula | added index |
| 4970 | used in formula | added index |
| 5486 | used in write to totdose.res | added index |
| 5493 | used in write to totdose.res | added index |

Changes to dcagw.f

| | | |
|-----------|--|---|
| 79 | passed to dcagw as argument | variable name change to designate the change to an array |
| 142 | double precision dilution(mxntime) | array added – this array is passed back to exec instead of variable dilutionvolume which is still used. |
| 1831-1837 | dilutvol() is set to either pluvdilutionvolume or dilutionvolume | logic added to store the dilution volume (either regular or pluvial) into new array which is |

Printed:

R. Rogers

SCIENTIFIC NOTEBOOK No. 170-18e

| | | |
|---|---|-----------------------------------|
| | | indexed on time steps. |
| 2018 | used in formula for $\text{remperyrpernuclgw}(j,k)$ | used instead of dilutionvolume |
| 2031 | same as 2018 | |
| 2120 | same as 2018 | |
| 2126, 2129, 2132, 2135, 2138, 2141 | same as 2018, but per pathway | |
| 2148 | same as 2018 | |
| 2154, 2157, 2160, 2163, 2166, 2169 | same as 2018, but per pathway | |

The dilution volume array, that is passed back from `dcagw()` to `exec()`, has only two distinct values – one for pluvial, the other for non-pluvial.

October 16, 2002 –

The test for SCR-399 should just be comparing `totdose.res` produced by old code vs. the new code. Check to see if there is a `totdos_c.res` file produced. This file is mentioned in the SCR write-up. `exec.f` outputs both `totdose.res` (`iunittotdose`) and `totdos_c.res` (`iunittotdose_c.res`). Note: only `totdose.res` is closed.

`totdose.res` – Total Dose for All Pathways, All Nuclides, and All Times

`totdos_c.res` – Total Dose for All Pathways, All Nuclides, for Compliance Period

October 30, 2002 –

Printed:

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

Verified with Ron that I would test SCR-384. Received spreadsheet from Mike Smith which he had used to check the formula in SCR-384.

November 5, 2002 –

Finished reviewing Mike Smith's spreadsheet for SCR-384 and corrected some errors. The values for ciper_{m2} will be compared against the values in the file ashrmovo.rlt.

Also need ashplume.out to verify the values of gramsashpercm₂ and gramsfpercm₂.

November 14, 2002

Helped Ron with diagnosing an error message NRC had gotten when running TPA under Windows 2000. The Occupational factors did not add up to 1.0.

November 21, 2002 –

Part of SCR-392 (DCAGW/DCAGS problem): Files should have been closed before exiting dcagw and dcags. Others believe the fix to this problem is with the variable inewdfdat, which is a unit number used to open the file "gnewdf.dat". Also, the file is never closed.

November 22, 2002 –

Make changes to dcagw.f and dcags.f to close the gnewdf.dat file, plus made corrections to dcags.f. Tested the change under Linux and sent it on to Ron to test it under Windows, where the problem occurs. [Side note: This problem was later found out to be caused by a lack of computer memory in the Windows computers.]

November 25, 2002 –

Working on problem with importance analysis. With ImportanceAnalysisFlag set to 1, iareader has problems reading ia.dat.

Printed:

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

Added a line after every occurrence of `backspace(iaunit)` in `iareader`, to decrement `ilinum` which keeps track of the number of lines read for `ia.dat`.

A call to `adduserdiscreteempirical(name,n,v)` in `sampler.f` will add a parameter to the array `description()` which `ispquery` uses to check and convert parameter names.

The code is currently failing in a call to `ispquery` with the argument `TSw_Thickness10SubArea[m]`. The error is a missing underscore after “Thickness” in the argument name. In `ia.dat`, the parameters for “10SubArea” are all missing a preceding underscore.

Ron mentioned that there may be problems when setting the `ImportanceAnalysis` flags associated with pumping rates.

November 27, 2002 –

Removed the `Fracture/Immobile` parameters from `ia.dat` that were not in `tpa.inp` and added the others.

After the changes, `tpa` fails when it runs `dcagw` (after completing subarea 10 in realization 1). As described elsewhere, `gentpa` is producing a negative leaching rate. Also described elsewhere, `dcagw` is having a problem with pumping rates.

December 13, 2002 –

NRC is having problems with the `ImportanceAnalysis` feature – they are getting negative leaching rates. They do not have the SCR-417 code fixes. See explanations in other sections.

January 31, 2003 –

For SCR-381, discovered problem causing questionable numbers for `Tc99` and `I129` and passed information on to Pat LaPlante.

Printed:

R. Rogers SCIENTIFIC NOTEBOOK No. 170-18e

Entries into pages 1-21 of this Scientific Notebook, part of #170-18e, have been made by Robert Rogers.

No original text entered into this Scientific Notebook has been removed.

Robert Rogers .

I have reviewed this scientific notebook and find it in compliance with QAP-001. There is sufficient information regarding methods used for conducting tests, acquiring and analyzing data so that another qualified individual could repeat the activity.

Paul 11-9-05