**CACIE Tool** – **ca-dosecalc**

**Version** **1.1**

**QA**: **QA**

# Description and Purpose

This tool calculates the dose for various exposure pathways at each point in the domain of a MODFLOW/MT3D model:

Given

1. A list of target contaminants of potential concern (COPCs) along with properties relevant to the dose calculation.
2. A list of exposure pathways
3. Groundwater concentrations in space and time for the target COPC in the form of MT3D output (a binary, .UCN file)
4. A shapefile defining the location of cells in the MODFLOW model grid
5. A comma-separated values (CSV) file relating a surface soil type to each row and column in the model
6. A CSV file relating COPC unit dose factors to exposure pathways and surface soil types

This utility creates a relational database, loads the above data sets and then:

1. Determines the unit dose factor corresponding to a target COPC for each exposure pathway at each grid cell.
2. Converts the MT3D concentrations (typically pCi/m3) to concentration units consistent with the unit dose factors (typically pCi/liter)
3. Calculates the dose vs time, space, and exposure pathway by multiplying the target COPC groundwater concentration by the corresponding unit dose factor for each valid combination of layer, row, column, time, surface soil type, and exposure pathway.
4. Writes the output to a comma-delimited file.

Ca-dosecalc is intended to be invoked by the CA/CIE Tool Runner once for every saturated zone model run.

# Functional Requirements

Ca-dosecalc has the following functional requirements (FR):

FR-1: Consume the MODFLOW grid shapefile work product

FR-2: Consume the unit dose factors work product

FR-3: Consume the surface soil type-row/column relationship work product

FR-4: Consume a binary .UCN file produced by a MODFLOW/MT3D model run

FR-5: Convert concentration units (unitsin) to concentration units (unitsout) consistent with unit dose factors using the conversion factor (conversion):  
 concentration (unitsout) = concentration (unitsin)\*conversion

FR-6: Compute the dose for each layer, row, column, time, and exposure pathway where the concentration is greater than or equal to the input threshold  
 dose = doseFactor\*concentration (unitsout)

FR-7: Output results to a CSV file

# Software Requirements Specifications

Ca-dosecalc is implemented as a series of linux shell scripts and SQL statements. The SQL is executed sequentially by the scripts to build the database, load the data as tables, compute the dose by building a materialized view, and exporting the results. This tool implicitly references python3 and the flopy python library and it also uses PostGIS to perform geospatial operations.

It is assumed that the tool will operate on a Linux (Ubuntu) machine with the following modules/libraries installed:

* PostgreSQL version 12+
* PostGIS version 2.4.3+
* shp2pgsql version 1.1.5
* Python 3.6+
  + numpy 1.19.0 +
  + pandas 1.0.5 +
  + flopy 3.3.1

# Software Design Description

This tool is intended to be invoked by the CA/CIE Tool Runner on a Linux machine. The command signature looks like:

$ python3 *[path to tools]*/tools/ca-dose/calcDose.py copc NLay gridShapefile ucnFile soilFile dosefactsFile copcFile pathwaysFile unitsin unitsout conversion startyear outputFormat modeldate outputFile

Arguments:

This tool expects the following arguments, all strings, each separated by a single space:

|  |  |  |
| --- | --- | --- |
| **Order** | **Argument Name** | **Argument Description** |
| 1 | Copc | The name of the target copc; ‘Tc99’, ‘U235’, etc. This must be present inside the *copcFile* |
| 2 | NLay | The number of layers in the MODFLOW grid |
| 3 | gridShapefile | Path to the shapfile (ending in .shp) defining the grid. |
| 4 | ucnFile | Path to the MT3D concentrations file (the .UCN file) |
| 5 | soilFile | Path to the CSV file containing the surface soil type to row-column relationships. |
| 6 | dosefactsFile | Path to the CSV file containing the unit dose factors for each COPC, surface soil type, and exposure pathway. |
| 7 | copcFile | Path to the CSV file containing a list of the target COPCs and COPC-specific information |
| 8 | pathwaysFile | Path to the CSV listing the exposure pathways |
| 9 | unitsin | Denotes the units of the concentrations in the UCN file; typically, ‘pCi/m^3’ |
| 10 | unitsout | Denotes the units of the concentration expected by the dose factors; typically, this is ‘pCi/L’ |
| 11 | conversion | Defined as the number that converts unitsin to unitsout:  unitsout = conversion\*unitsin |
| 12 | startyear | The beginning year of the MT3D simulation |
| 13 | outputFormat | A formatting string, Postgres-style, that denotes the format of the dose output. Typically, this is '9.99999999EEEE’ |
| 14 | modeldate | The date that the UCN file was produced in the format ‘YYYY-MM-DD’. This is currently only used as an internal label; any valid string is acceptable. |
| 15 | outputFile | Path to the output CSV file and file name. |

|  |  |
| --- | --- |
|  |  |

Input Files:

This tool will malfunction if input files are not formatted as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Input File** | **File Format** | **Typical extension** | **Details** |
| gridShapefile | ESRI Shapefile | .shp | The ICF-controlled MFGRID work product containing the P2R v 8.3 MODFLOW model grid extent |
| soilFile | CSV | .csv | The ICF-controlled SOILIND work product containing the soil index for each row and column. This is a CSV file with a header row and columns:  ID (string; ignored)  ROW (integer)  COL (integer)  SOIL\_CATEGORY (string)  SOIL\_INDEX (integer); must be identical to an element in the SOIL\_INDEX column of the dosefactsFile. |
| ucnFile | MT3D | .ucn | A valid MT3D binary output containing concentrations. |
| copcFile | CSV | .csv | A valid CSV without a header containing eight columns:   1. The number 1 2. The copc short name (e.g. Tc99). This must be identical to a value in the dosefactsFile 3. The copc short name, identical to the text in column 2. 4. The copc short name, identical to the text in column 2 and column 3. 5. The mcl for this COPC as a float (ignored by this tool) 6. The number 3 7. The cutoff threshold, as a float (e.g 1e-6). This forces the concentration to exactly zero, “0” for anything below this value. Units must be identical to those in the ucnFile. Setting a high threshold greatly improves efficiency since dose is only calculated for non-zero concentrations. 8. The copc type as a string: “rad” or “chem”. This parameter is currently ignored. |
| dosefactsFile | CSV | .csv | A valid CSV. The first line must be a header and have the following column names: SOIL\_INDEX, SOIL\_CATEGORY, COPC, Pathway, Dose Factor. The columns must contain the following:   1. SOIL\_INDEX must be identical to a soil index in soilFile; this is used by the tool to relate soil categories in this file to those in soilFile. 2. SOIL\_CATEGORY must be a string; this is ignored by the tool and the category name is obtained from soilFile 3. COPC must be the short COPC name (e.g. Tc99) and must be identical to the copc names in the copcFile. 4. Pathway must be a string in the pathwaysFile 5. Dose Factor must be a float denoting the dose factor with dimensions [dose units]/[concentration units]. The dose units determine the units of the dose calculated in the output file. The concentration units must match the concentration units in ucnFile OR be equal to conversion\*unitsin. |
| pathwaysFile | CSV | .csv | A valid CSV without a header.  First column must be the number 1, second and third columns must be identical and contain the name of the exposure pathway.  Exposure pathway names must be identical to those in dosefactsFile. |

Output Files:

This tool produces the following output files:

TestReport.txt file which logs the processing date, time, and steps of the tool.

CSV file containing the calculated dose for each exposure pathway at every point in time and space in the model domain. Each row represents a unique space/time/pathway. Path and file name is provided as an argument to the tool.

The first row is header text with column names. Columns correspond to:

|  |  |  |  |
| --- | --- | --- | --- |
| **Index** | **Column name** | **Type** | **Description** |
| 1 | elapsed\_tm | Integer | The number of days since the start of the model corresponding to this row’s data |
| 2 | model\_date | Text | The calendar date in YYYY-MM-DD format corresponding to elapsed\_tm. The calendar date is calculated as the model start date (determined by the modeldate input parameter) plus elapsed\_tm calendar days. Leap years are observed. |
| 3 | soil | Text | The soil type of the grid cell (row/col) |
| 4 | pathway | Text | The exposure pathway |
| 5 | cell\_row | Integer | grid index (1-based) |
| 6 | cell\_column | Integer | grid index (1-based) |
| 7 | cell\_layer | Integer | grid index (1-based) |
| 8 | concentration | Float | The concentration, in units of the MT3D UCN file, multiplied by the conversion parameter. |
| 9 | dose\_factor | Float | The dose factor for the pathway/soil type  Units are the same as those provided in dosefactsFile |
| 10 | dose | Float | The calculated dose. Equivalent to column 8 \* column 9. |

File size depends strongly on the number of grid cells, timesteps, and exposure pathways; for the output produced by saturated zone model, output is typically on the order of 8GB.

Tool Runner:

The following is the shell script configuration that will be passed as an argument to the Tool Runner for qualified runs:

$ python3 *[path to tools]*/tools/ca-dose/calcDose.py copc NLay gridShapefile ucnFile soilFile dosefactsFile copcFile pathwaysFile unitsin unitsout conversion startyear outputFormat modeldate outputFile

Code Review:

Version 1.0 code walkthrough was performed by Neil Powers on 07/20/2020. The code relies on bash shell files to execute POSTGRES SQL to generate the database. No impacts to other repository tools or shared library dependencies were identified for the ca-dosecalc.

Version 1.1 code walkthrough was performed by Neil Powers on 8/10/2020; there were no additional comments.

# Requirements Traceability Matrix

The requirements traceability matrix for the ca-dosecalc tool is presented in Table 1.

| Table 1  Requirements Traceability Matrix | | |
| --- | --- | --- |
| **Functional Requirement ID** | **Acceptance Test ID** | **Test Case – step** |
| QA Level | CACIE-ca-dosecalc-AT-1 | 1-1: Tool was invoked and executed |
| FR-1 | CACIE- ca-dosecalc-AT-1 | 1-3: Verify MODFLOW grid shapefile processed by checking testReport.txt |
| FR-2 | CACIE-ca-dosecalc-AT-1 | 1-3: Verify unit dose factors file processed by checking testReport.txt |
| FR-3 | CACIE-ca-dosecalc-AT-1 | 1-3: Verify surface soil type-row/column file processed by checking testReport.txt |
| FR-4 | CACIE-ca-dosecalc-AT-1 | 1-3: Verify binary .UCN file processed by checking testReport.txt |
| FR-5 | CACIE-ca-dosecalc-AT-1 | 1-6.3.3: Verify concentration is converted using conversion factor specified in argument list |
| FR-6 | CACIE-ca-dosecalc-AT-1 | 1-6.1: Determine conversion factor provided as argument  1-6.2: Verify output file unit dose factor is consistent with dose factor input file relative to COPC, pathway and soil type  1-6.3: Verify output file groundwater concentration is consistent with UCN input file relative to COPC, time and grid cell (layer, row, column)  1-7: Verify dose calculation is correct  1-9: Verify dose calculation is performed for groundwater concentrations greater than COPC-specific threshold value |
| FR-7 | CACIE-ca-dosecalc-AT-1 | 1-2, 1-3, 1-4, 1-5: Verify results saved to a .csv file consistent with the formatting specified in Section 4 |

# Installation Test Plan and Acceptance Test Plan Cases

This tool will be tested on the machine that it will be used on. The installation test and the acceptance tests are therefore the same test and presented in Table 3.

| Table 3  **ca-dosecalc Acceptance Test Plan Case 1** | | | |
| --- | --- | --- | --- |
| **ca-dosecalc Acceptance Testing**  **CACIE-ca-dosecalc – AT-1** | | **Date:** | |
| **Tool Runner Log File Location for this test:**  **~/dose/testDose/test/output/runlog.txt** | | **Test Performed By:** | |
| **Testing Directory: ~/dose/testDose** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Remote into the machine holding the tool as the “ca” user using the password provided by the developer  Navigate to the Testing Directory | | | |
| 1 | Run the script by typing ‘./runAT1.sh’ into the command shell and let the program run to completion.  *Note that the program may take a half-hour to complete. When the program is finished, the shell will display the command prompt.* | The program begins to execute  The shell may display text such as “Field delx is an FTDouble “ and various “Notices”. |  |
| 2 | When the program finishes,  navigate to the output directory by typing cd ~/dose/testDose/test/output and pressing enter  type ls -alh and press enter | The output directory contains four files:   * runlog.txt * testReport.txt * U236.csv * U236-conc.csv |  |
| 3 | Open testReport.txt and verify that the following text is present:   * Finished loading grid shapefile * updated cells: added layer X [where X ranges from 1 – 7] * END loading soils map * Loaded pathways * Loaded copcs * Loaded dose factors * END; Loaded concentration table for … * END Export of dose data | The required text is present in the output file. |  |
| 4 | Type head U236.csv and press enter. Examine the first line of the output. | Text displays in the console window. The first line is a list of column names separated by commas. The column names should be, in order:   * elapsed\_tm * model\_date * soil * pathway * cell\_row * cell\_column * cell\_layer * concentration * dose\_factor * dose |  |
| 5 | Again, type head U236.csv and press enter. Examine the second line of the output. | This line contains primarily numeric data separated by commas:   * the first column is the integer 365 * the second column is a date in the format ‘YYYY-MM-DD’. The date is 2019-01-01. * the third column is a soil type (Rupert Sand, Burbank Loamy, or similar) * the fourth column is a pathway: Soil, Egg, Drinking Water, or similar * the fifth, sixth, and seventh columns are integers * the eighth and ninth columns are floats |  |
| 6.1 | cd to the test directory by typing cd ..  and pressing Enter  Open the input script (runAT1.sh) and write down the number in the line containing the text conversion= | The conversion number is: | |
| 6.2.1 | Select a row of data in output file | Selected row: | |
| 6.2.2 | Record pathway and soil type corresponding to selected row | Pathway:  Soil Type: | |
| 6.2.3 | Record target COPC (argument provided to tool per Section 4) | Target COPC: | |
| 6.2.4 | Verify unit dose factor listed in selected row of the output file is consistent with unit dose factor listed in input file (home/ca/dose/testDose/test/inputs/Soil\_Specific\_UDF\_CA\_Model\_flat.csv) relative to target COPC and the soil type and pathway. | Output unit dose factor =  Corresponding dose factor in input file |  |
| 6.3.1 | Run the following checking utility program by entering the following command (in the testing directory):  **python3** checkUcn.py | | |
| 6.3.2 | Verify time, layer, row, and column of utility output is consistent with time, layer, row, and column of output file | Utility program output time, layer, row, column, values = time, layer, row, column values of first row of data in the U236.csv file |  |
| 6.3.3 | Verify target COPC groundwater concentration in output file (U236.csv) is consistent (conversion factor considered) with UNC input file relative to year and grid cell (layer, row, column). You may need to search the file. | Output groundwater concentration and conversion in the matching row of the U236.csv file equals utility tool’s reported groundwater concentration |  |
| 7 | Verify dose calculation as follows (or use Excel or equivalent program):  cd to the output directory by typing cd .. and pressing Enter  type head U236.csv into the console and press Enter.  For any row except the first row, verify that column 10 = column 8 \* column 9 | Column 10 = column 8 \* column 9 |  |
| 8 | Type cd ../inputs and press Enter  Open copcs.csv  Find the line containing U236 and write down the number in the seventh column and multiply it by the conversion factor. This is the COPC-specific threshold | The number is = | |
| 9 | Check that dose was calculated for concentrations greater than the COPC-specific threshold as follows (or use Excel or equivalent program):  Type cd ../output and press Enter  Type head U236.csv and press Enter  Verify that all of the entries in the eighth column are greater than or equal to the number in step 8 | All the numbers in column eight (concentration) are greater than or equal to threshold |  |

# Acceptance Test Report

To complete the Acceptance Testing use Appendix A. Acceptance Test 1 is in Table A-1.

Details of these tests, when they were conducted, by whom, and if they Passed or Failed are in each table of Appendix A.

# User Guide

Ca-dosecalc is intended to be invoked on the command line by the CA/CIE tool runner. We recommend the following operating procedure for each COPC:

1. Assemble the relevant input files

2. Create a run script. If you are processing multiple UCN files, take care to ensure that the COPC is defined correctly, that the output file name is unique, and that there is enough space on the machine to store the output. Output file sizes depend strongly on the number of exposure pathways, grid size, and timesteps. Experience has shown that with the MFGRID v 8.3 and threshold around 1e-6, output file sizes are typically around 8GB each.

An example run script is provided below:

|  |  |
| --- | --- |
|  | #--------------------------------------------------------------------- # Example ca-dose usage script |
|  | # |
|  | NLay=7 |
|  | gridShapefile='/home/ca/dosecalc/source/MFGRID/v8.3/data/grid\_274\_geo.shp' |
|  | soilFile='/home/ca/dosecalc/source/SOILIND/v1.0/data/mfgrid\_soil\_indices.csv' |
|  | ucnFile='/home/ca/dosecalc/source/testConc/u235/P2RGWM.ucn' |
|  | copc='U235' |
|  | unitsin='pCi/m^3' |
|  | unitsout='pCi/m^3' |
|  | conversion='1' |
|  | startyear='1944' |
|  | modeldate='2020-07-13' |
|  | outputFormat='9.99999999EEEE' |
|  |  |
|  | doseFiles='/home/ca/dosecalc/dev/fakeDoseFactorData/' |
|  | pathwaysFile=$doseFiles'pathways.csv' |
|  | copcFile=$doseFiles'tempcopc.csv' |
|  | dosefactsFile=$doseFiles'tempdose.csv' |
|  | outputFile=$doseFiles'output/'$copc'.csv' |
|  | toolsDir='/home/ca/dosecalc/CA-CIE-Tools' |
|  | cmd=$toolsDir'/tools/ca-dosecalc/calcDose.py' args=”$cmd $copc $NLay $gridShapefile $ucnFile $soilFile $dosefactsFile $copcFile $pathwaysFile $unitsin $unitsout $conversion $startyear $outputFormat $modeldate $outputFile" toolRunner=’home/ca/dosecalc/CA-CIE-Tools/pylib/runner/runner.py’  python3 $toolRunner --logfile “toolRunnerReport.txt” “python3” “$args” > runlog.txt #------------------------------------------------------ end of example script |

3. Ensure your script has execute privileges

4. Run your script.

Run times may be a half-hour or longer. We recommend monitoring progress and resource consumption with htop.

# Tool Versions

This section details changes incorporated into each version of the **ca-dosecalc** tool.

* 1.0 – Tool was developed.
* 1.1 – Submitted 8/10/2020, this version addresses a bug in pop\_concentrations.sh that was causing cells to be mislabeled.

# Appendix A

**Completed Acceptance Test Cases**

**Tool Runner Log**

INFO--08/10/2020 04:25:33 PM--Starting CA-CIE Tool Runner. Logging to "output/runlog.txt"

INFO--08/10/2020 04:25:33 PM--Code Version: 63140375918466bd0d783e985d6fe8e0eda338f4 v4.0: /home/ca/CA-CIE-Tools/pylib/runner/runner.py<--1bcfd6779e9cbdb82673405873a8e5e81514ae27

INFO--08/10/2020 04:25:33 PM--Code Version: 4b4bee3be9752f02f6524dc35edf8c6ecf2ccc5b Local repo SHA-1 has does not correspond to a remote repo release version: /home/ca/dose/testDose/test/CA-CIE-Tools/tools/ca-dosecalc/calcDose.py<--1cc4d017c7afdad176e4f6df077aad74b76a3398

INFO--08/10/2020 04:25:33 PM--QA Status: QUALIFIED : /home/ca/CA-CIE-Tools/pylib/runner/runner.py

INFO--08/10/2020 04:25:33 PM--QA Status: TEST : /home/ca/dose/testDose/test/CA-CIE-Tools/tools/ca-dosecalc/calcDose.py

INFO--08/10/2020 04:25:33 PM--Invoking Command:"python3" with Arguments:"/home/ca/dose/testDose/test/CA-CIE-Tools/tools/ca-dosecalc/calcDose.py U236 7 /home/ca/dose/testDose/test/inputs/MFGRID/v8.3/data/grid\_274\_geo.shp /home/ca/dose/testDose/test/inputs/u236/P2RGWM.ucn /home/ca/dose/testDose/test/inputs/SOILIND/v1.0/data/mfgrid\_soil\_indices.csv /home/ca/dose/testDose/test/inputs/Soil\_Specific\_UDF\_CA\_Model\_flat.csv /home/ca/dose/testDose/test/inputs/copcs.csv /home/ca/dose/testDose/test/inputs/pathways.csv pCi/m^3 pCi/L 0.001 2018 9.99999999EEEE 2020-07-20 /home/ca/dose/testDose/test/output/U236.csv"

INFO--08/10/2020 04:25:33 PM--Username:ca Computer:twotbbase Platform:Linux 4.15.0-111-generic #112-Ubuntu SMP Thu Jul 9 20:32:34 UTC 2020

| Table 3  **ca-dosecalc Acceptance Test Plan Case 1** | | | |
| --- | --- | --- | --- |
| **ca-dosecalc Acceptance Testing**  **CACIE-ca-dosecalc – AT-1** | | **Date:** 8/10/2020 | |
| **Tool Runner Log File Location for this test:**  **~/dose/testDose/test/output/runlog.txt** | | **Test Performed By:** Christian Hall | |
| **Testing Directory: ~/dose/testDose** | | | |
| **Test Step** | **Test Instruction** | **Expected Result** | **Test Result  (Pass/Fail)** |
| Remote into the machine holding the tool as the “ca” user using the password provided by the developer  Navigate to the Testing Directory | | | |
| 1 | Run the script by typing ‘./runAT1.sh’ into the command shell and let the program run to completion.  *Note that the program may take a half-hour to complete. When the program is finished, the shell will display the command prompt.* | The program begins to execute  The shell may display text such as “Field delx is an FTDouble “ and various “Notices”. | PASS |
| 2 | When the program finishes,  navigate to the output directory by typing cd ~/dose/testDose/test/output and pressing enter  type ls -alh and press enter | The output directory contains four files:   * runlog.txt * testReport.txt * U236.csv * U236-conc.csv | PASS |
| 3 | Open testReport.txt and verify that the following text is present:   * Finished loading grid shapefile * updated cells: added layer X [where X ranges from 1 – 7] * END loading soils map * Loaded pathways * Loaded copcs * Loaded dose factors * END; Loaded concentration table for … * END Export of dose data | The required text is present in the output file. | PASS |
| 4 | Type head U236.csv and press enter. Examine the first line of the output. | Text displays in the console window. The first line is a list of column names separated by commas. The column names should be, in order:   * elapsed\_tm * model\_date * soil * pathway * cell\_row * cell\_column * cell\_layer * concentration * dose\_factor * dose | PASS |
| 5 | Again, type head U236.csv and press enter. Examine the second line of the output. | This line contains primarily numeric data separated by commas:   * the first column is the integer 365 * the second column is a date in the format ‘YYYY-MM-DD’. The date is 2019-01-01. * the third column is a soil type (Rupert Sand, Burbank Loamy, or similar) * the fourth column is a pathway: Soil, Egg, Drinking Water, or similar * the fifth, sixth, and seventh columns are integers * the eighth and ninth columns are floats | PASS |
| 6.1 | cd to the test directory by typing cd ..  and pressing Enter  Open the input script (runAT1.sh) and write down the number in the line containing the text conversion= | The conversion number is: 0.001 | |
| 6.2.1 | Select a row of data in output file | Selected row: 3 | |
| 6.2.2 | Record pathway and soil type corresponding to selected row | Pathway: Drinking Water  Soil Type: Ephrata Sandy Loam | |
| 6.2.3 | Record target COPC (argument provided to tool per Section 4) | Target COPC: U236 | |
| 6.2.4 | Verify unit dose factor listed in selected row of the output file is consistent with unit dose factor listed in input file (home/ca/dose/testDose/test/inputs/Soil\_Specific\_UDF\_CA\_Model\_flat.csv) relative to target COPC and the soil type and pathway.  ***Tester’s Note:*** Dose factor of 0.188 | Output unit dose factor =  Corresponding dose factor in input file | PASS |
| 6.3.1 | Run the following checking utility program by entering the following command (in the testing directory):  **python3** checkUcn.py | | |
| 6.3.2 | Verify time, layer, row, and column of utility output is consistent with time, layer, row, and column of output file  ***Tester’s Note:***  Output: checkUcn.py  Outfile file: U236.csv  Lines Numbers:  Beef: 13677  Drinking Water: 13692  Egg: 13707  External Gamma: 13722  Inhalation: 13737  Milk: 13752  Poultry: 13767  Produce: 13782  Soil: 13797  Total: 13812 | Utility program output time, layer, row, column, values = time, layer, row, column values of first row of data in the U236.csv file | PASS |
| 6.3.3 | Verify target COPC groundwater concentration in output file (U236.csv) is consistent (conversion factor considered) with UNC input file relative to year and grid cell (layer, row, column). You may need to search the file.  ***Tester’s Note:*** Since the ucn script outputs time, layer, row, col, and concentration, it was easiest to find the timestamp in the output file, and then narrow in on the correct line by using “row,col,layer”.  Line Number in U236.csv: 6692 | Output groundwater concentration and conversion in the matching row of the U236.csv file equals utility tool’s reported groundwater concentration | PASS |
| 7 | Verify dose calculation as follows (or use Excel or equivalent program):  cd to the output directory by typing cd .. and pressing Enter  type head U236.csv into the console and press Enter.  For any row except the first row, verify that column 10 = column 8 \* column 9  ***Tester’s Note:*** (Line 1) 1.356 \* 0.000496 = 0.0006728 | Column 10 = column 8 \* column 9 | PASS |
| 8 | Type cd ../inputs and press Enter  Open copcs.csv  Find the line containing U236 and write down the number in the seventh column and multiply it by the conversion factor. This is the COPC-specific threshold  ***Tester’s Note:*** 800.0 \* 0.001 = 0.8 | The number is = 0.8 | |
| 9 | Check that dose was calculated for concentrations greater than the COPC-specific threshold as follows (or use Excel or equivalent program):  Type cd ../output and press Enter  Type head U236.csv and press Enter  Verify that all of the entries in the eighth column are greater than or equal to the number in step 8 | All the numbers in column eight (concentration) are greater than or equal to threshold | PASS |