# AGNPS VERSION 5.00 VERIFICATION REPORT: Software

John Witte, Fred D. Theurer, & Kevin D. Baker

This document describes the system changes and additions made to the AGNPS 5.00 computer program for the science module AGRUN.EXE. The non-science modules (input/output routines such as the spreadsheet editor) changes were limited to accepting version 5.00 in the header line in the data file and display for the screen ID's.

The origin for AGRUN version 5.00 was AGRUN version 4.03b. It was coded in the C programming language using Borland C++ version 3.0 DOS compiler and the Unix ANSI-Standard compiler on a Sun-LX Workstation. Version 4.03b was more complete than the AGRUN 4.03 which was the previous official version released by ARS in October of 1994. AGNPS user responses to the last release of AGNPS 4.03 prompted this verification, revision, and release of AGRUN version 5.00.

## ADDITIONAL FILES ADDED TO 5.00

The AGRUN version 5.00 verification process required additional outputs to be created during runtime of the model. The outputs are user-specified flags which are classified into basic debug flags, routine debug flags, and verification table flags. These flags may be set in the DEBUG.FLG file using a text editor and will be explained later in this document. The debug information (basic and routine) are written to the screen while AGRUN version 5.00 is running. The output tables were divided into the three main processes which were verified, that is: Water, Sediment, and Chemicals. These tables are also user-specified in the DEBUG.FLG file and create separate output files during run-time. Several routines were added to AGRUN 5.00 for this verification enhancement. These routines were added in the DEBUGFLG.C source file and the DEBUGFLG.H include file which are new for the AGRUN version 5.00 model.

### Discussion

The addition of these files to the source code required additional include file definitions in each of the source (*.C) files as well as structure declarations within the routines where the new data structures were used. The definitions were added using proper ANSI syntax to retain portability to UNIX platforms. These files were also added to the Borland AGRUN500.PRJ project file and the Phar Lap compilation batch input file AGRUN500.RSP.

## DESCRIPTION OF DEBUG ADDITIONS

**Table 1: Basic Debug Flags**

| Flags | Description |
|---|---|
| Drain_area | Cell's drainage area calculations. |
| Cell_info | Cell position and characteristics. |
| Channel | Channel geometry values. |
| Flow | Flow values such as volume and peak. |
| Impound | Impoundment output calculation values. |
| Nutrient | Soil nutrient values in runoff. |
| Routing | Cell routing information. |
| Sediment | Sediment particle information. |
| TR-55 | TR-55 method of overland flow calculations. |

The debug additions in the verification process for AGRUN version 5.00 are specified by Basic and Routine flags. As mentioned earlier, these are user-specified flags found in the DEBUG.FLG file. The debug flags file lists each of these categories along with the specific variable needed for debugging followed by a "0" or a "1". Values marked with a "0" are turned off while those marked with a "1" are turned on. The values which are turned on are then printed out to the screen during run-time of the AGRUN version 5.00 model. The screen output may be captured by redirecting the output to a file when the model is run as follows:

AGRUN [*watershed name.DAT*] 0 0 0 > [*screen capture filename*]

The output of the screen capture file may then be viewed after the AGRUN execution using a text editor. Table 1 shows the basic debug flags and Table 2 shows the routine debug flags.

## Discussion

These values are read into the FLAGS_BASIC and FLAGS_ROUTINE data structures which are defined in the DEBUGFLG.H file. The call to the file to read these flag values is made from the AGRUN.C source file. Note that a check has been added to the main function in AGRUN.C which turns off the debug and verification flags if more than a nine cell watershed is being run due to verification table formatting problems. This limitation will be eliminated in the design and implementation of the continuous simulation AGNPS version (ANNAGNPS).

**Table 2: Routine Debug Flags**

| Flag | Description |
|---|---|
| Bulk_dens | Soil bulk density lookup table. |
| Calc_chan | Channel parameters calculated for TR-55. |
| Calc_sed | Sum of sediment values for the cell. |
| Cell_calc | Calculated cell values. |
| Chan_calc | Channel parameters calculated for AGNPS. |
| Chan_TR-55 | Actual TR-55 calculations. |
| Curve_num | Calculates retention value and runoff volume. |
| Curve#_ro | SCS curve number method from NEH-4. |
| Decay_nut | Decay nutrients within the channel. |
| Feedlot | Feedlot submodel calculations. |
| Hydrology | Calculates peak flow and duration as CREAMS. |
| Impound | Sets impoundment parameters (including sediment). |
| Init_sed | Set up sediment by particle class. |
| Length | Channel parameter calculations. |
| Newsoil | Calculates cell sediment deposition. |
| Overland | Calculates time needed for concentrated flow. |
| Part_pest | Partitions pesticide on sediment and in water. |
| Pesticide | Pesticide calculation routine. |
| Pntsrc | Sums up feedlot model output. |
| Prep_route | Longest flow path to current cell. |
| Rclmap | Receiving cell mapping routine. |
| Recursive | Recursive routing function. |
| Routing | Actual routing and accumulation. |
| Sed_flow | Sediment flow and transport capacity for cell. |
| Sed_nutr | Calculate nutrients in sediment. |
| Soil_nutr | Calculate nutrients in soil. |
| Src_cells | Finds and marks source (primary) cells. |
| Sum_Psrc | Sums point source nutrients. |
| Terh | Calculate peak flow from terrace impoundment. |
| Ters | Calculates sediment yield from a terrace impoundment. |
| TR-55 | Calculates overland flow TR-55 method. |
| Xeros | Soil erosion based on USLE. |

# DESCRIPTION OF VERIFICATION TABLES

Verification tables were added in AGRUN version 5.00. They are specified by Water, Sediment, and Chemicals. As mentioned earlier, these are user-specified flags found in the DEBUG.FLG file. The debug flags file lists each of these categories along with the specific variable needed for debugging followed by a “0” or a “1”. Values marked with a “0” are turned off while those marked with a “1” are turned on. *Note that there are three other processes listed in the DEBUG.FLG file verification table for feedlots, impoundments, and sediment traps. These debug features were not implemented for this release.*

The requested verification table values which are turned on are then written to a formatted file: (a) Water to *.vt1; (b) Sediment to *.vt2; (c) Chemicals to *.vt3 *(where * represents the watershed data file name).* The files are formatted to be imported into MS-Word for Windows version 6.0c. Once the file is imported into MS-Word, change the Page Setup (FILE menu) to landscape printout type and set the document font type to *LinePrinter*. The formatted files may then be printed on letter-sized paper (8.5” x 11”). See Appendices A, B, & C for glossary tables of each. A similar procedure can be used for WordPerfect users. Table 3 shows the verification table flags.

### Discussion

**Table 3: Verification Table Flags**

| Flags | Description |
|---|---|
| Water_Table | Water process tabular output. |
| Sed_Table | Sediment process tabular output. |
| Chem_Table | Chemical process tabular output. |
| Fdlt_Tab | Feedlot process tabular output. < Not Implemented > |
| Impnd_Table | Impoundment tabular output. < Not Implemented > |
| Sdtrp_Table | Sediment trap tabular output. < Not Implemented > |

These values are read into the FLAGS_TABLE data structures which are defined in the DEBUGFLG.H file. The call to the file to read these flag values is made from the AGRUN.C source file. *Note that a check has been added to the main function in AGRUN.C which turns off the debug and verification flags if more than a nine cell watershed is being run due to verification table formatting problems. This limitation will be eliminated in the design and implementation of the continuous simulation AGNPS release.* The routines which create the table first write a mask file before model calculations are performed. The values in the selected data structures are assigned during the model calculations. When the model calculations are completed, the results are dumped to the *filename*.vt1, *filename*.vt2, or *filename*.vt3 data file for each undivided cell or column using the mask file. As mentioned above, these tables have been formatted to use the Windows version of MicroSoft Word version 6.0c. To print out a hard copy of the verification output tables, follow these instructions:

1. Begin MicroSoft Word for Windows version 6.0c
2. Select the File - Open menu item or click the Open File icon
3. Select the *.vt? file to be viewed and click the OK button
4. Once the file is loaded, select the File - Page Setup menu item
5. Next, select the Paper Size selection box and make sure the Orientation type is set to Landscape (in the lower-left corner of the selection box).
6. Using the mouse, select the entire document and select the LinePrinter font type.
7. Print out by selecting the printer icon in the menu bar.

The requested verification table values which are turned on are then written to your active print device.

## DESCRIPTION OF DEBUG.FLG

The DEBUG.FLG input data file was created specifically for the AGRUN 5.00 verification process but will act as a prototype for the design of the ANNAGNPS model. The file is necessary only for verification and need not be present for the AGRUN 5.00 model to run. If the file is present in the executable directory, it will be read and checked for any flag values which are “turned on”. *Note that the file must be formatted as specified in the Debug Flag File specifications. A sample of this file will be sent out with the release to act as a template.* The file is in ASCII format and may be updated using any text editor. *Note that a check has been added to the main function in AGRUN.C which turns off the debug and verification flags if more than a nine cell watershed is being run. This decision was made to avoid verification table coding complexities. This limitation will be eliminated in the design and implementation of the ANNAGNPS release.*

### Discussion

This file option was added to fulfill the need for additional output ability without the need for re-compilation. By adding a file check before any reading takes place, this addition will be completely invisible to the common AGNPS user. This exercise also will be used to prototype a similar method of flag manipulation which will take place in the continuous simulation AGNPS model. Some current uses of this feature is to assist the support staff in error tracking for version 5.00.

## CHANGES MADE TO SOURCE CODE FOR VERSION 5.00

The changes made to the source code during the verification process are divided into two categories: (a) science (the physical process models); and (b) software (the computer program). The science changes are addressed in a separate document named “AGNPS Version 5.00 Verification: Science” while this document addresses the changes made to the software (code) in version 5.00.

## Discussion

The verification additions required obvious additions to the source code. Each of the verification tables created required the addition of a data structure to store the respective values. These values were assigned in the source code as the calculations occurred. The initialization and the output of these data structures were all added in the DEBUGFLG.H and DEBUGFLG.C files, while the actual data structure assignments were added throughout the model source code. To include these assignments in the model source code, the addition of the DEBUGFLG.H include file was added to each *.C file. Any file which contained such assignments required the declaration of the appropriate data structure. Following these changes to the file headers, the actual assignments of the values were added to the source code as well. Each assignment was preceded by a flag check in order to not violate the rules of protected mode programming. To further illustrate the software additions, here is a section of added code taken directly from the AGRUN 5.00 version of the SEDNUT.C program:

```
#include "debugflg.h"
extern FLAGS_TABLE tflags;
extern CHEM_TABLE   ctable;
if (tflags.chem_table)
  {
    ctable.ct[column].N_tot_enrichment = nit_enrichment_ratio;
    ctable.ct[column].P_tot_enrichment = phos_enrichment_ratio;
  }
```

The first line is the additional include statement required for version 5.00. The second line describes the structure declaration necessary for interpreting the desired flags to be printed out. The third line is the declaration of the chemical verification table data structure. This is the actual data structure to be printed to the *.VT3 file. The last several lines illustrate the method used for assignments of the new data structures. Note that each of the data elements are indexed by column as they are in the output verification tables.

The Phar Lap 286|Dos-Extender Run-Time Kit (RTK) is used in AGRUN versions 4.0 and above. While there were no literal changes required in the source code to use Phar Lap, several coding structures were changed to fit the needs of the compiler. The most common problem found was violations of protected mode programming specifications. Such violations would cause the following error message:

```
Fatal Error 286.3330: General protection fault Detected.
```

This error message would then be followed by a register dump to the screen. Such items which were corrected to avoid such errors included NULL pointer checks and memory allocation changes. If any function used a NULL pointer for any reason, such a violation would occur. Therefore, any file closings and pointer manipulation must be checked for the NULL pointer possibility. Also, large watersheds require the use of more than 64K memory block allocations. For this reason, a FAR memory call was added to the READINPU.C file main data structure memory allocation command. This allowed the protected mode memory segment to use correct blocks of memory as needed. The compilation method of using the Phar Lap RTK will be described in detail in the “Compilation Method for AGRUN5.00” section later in this document.

The upgrade version 5.00 involved simple version print and read statement modification. These changes were placed in the “READINPU.C” source file and allows messages corresponding to the correct version of AGRUN which is version 5.00.

# OTHER POTENTIAL RUN-TIME ERRORS IN 5.00

Some other potential run-time errors were found in the current released version of AGRUN 5.00, but did not occur during any of the verification tests. These errors were noted, but not changed for this release. However, the continuous simulation AGNPS model will correct these potential problems. The vast majority of these problems are Floating point or Divide by 0.0 errors. These errors occur when an unchecked value is divided by a 0-valued variable. Such a situation will cause the program to print out an error message and then crash or lock up.

Another problem which was not addressed completely in this release deals with the spreadsheet interface (AGNPS.EXE) program. Two instances have been located in which values are not updated correctly when changes are made in the spreadsheet program. One such place this happens is in the default values from Geomorphic to Non-Geomorphic calculation methods, the other is in the Pesticide input screen when moving from post to pre-emergence application methods.

### Discussion

While the implications of leaving these values unchecked leaves more responsibility to the end user of the model, the time requirements of adding error trapping features to the single-event model would take away from the completion of the continuous simulation AGNPS model. The potential errors most likely to cause problems have been corrected and the remaining errors will only be exposed in extreme run situations. As a rule, if a Floating Point Divide by 0.0 Error is encountered, check the channel geometry inputs for 0.0 values as this is the where many of the unchecked equations are located within the model.

## MAXIMUM NUMBER OF CELLS FOR IBM-COMPATIBLES

Tests have been run on Gateway, AST, & Toshiba IBM-compatible systems with 80486 processors along with a Dell PC with a pentium processor. The machines range in memory sizes from 8 MB of RAM to 16 MB of RAM. The testing method used the fractal programs to generate square fractal watershed in sizes from a single cell to a 200x200 cell watershed containing 40,000 homogenous cells. Each cell was identical. They contained only the basic data necessary (no feedlots, water cells, etc.) and resulted in the minimum input data file size for the total number of cells used to define the watershed. Any cell input data more than these basic requirements (the inclusion of feedlots, etc.) will create a larger input file size for the same number of cells. *Note that this section describes the AGRUN.EXE main model itself. Other memory managers are used for the input and output interface programs, therefore large watersheds should be run from outside the spreadsheet (AGNPS.EXE) program. To run the model apart from the interface, type the following command from the AGNPS 5.00 executable directory once the *.DAT [watershed input file] has been correctly formatted:*

AGRUN [*watershed name.DAT*] 0 0 0

To date, the most confident maximum tested cell size is 16,384 or 128 x 128 square watershed. *Note that a fractal watershed of 32,761 cells or 181 x 181 was successfully run on the UNIX operating system.*

### Discussion

While 16,384 generated cells have been run under AGRUN version 5.00, it is important to note that this value is machine-dependent as well as data-dependent. The extended memory manager (Phar Lap) uses available Random Access Memory (RAM) before using Virtual Memory (hard disk swapping). If little RAM is available on a machine and/or little free disk space is available, a substantially smaller amount of cells will encounter "Out of Memory" errors. The test watersheds also used cells without a large amount of data. The dynamic nature of the AGRUN version 5.00 design will allow several more basic cells to run than cells with several added values within such as feedlots, pesticide applications, or impoundments. Therefore, if running large watershed on relatively small hardware, consider using a minimum amount of cell input.

## COMPILATION METHOD FOR DOS VERSION AGRUN 5.00

The various AGNPS components (i.e. model, spreadsheet, grafix, etc.) have been compiled using different methods and different software. For instance the science (model), and check utility are written in the C programming language and compiled using the Borland C++ version 3.0 compiler. The remaining utilities, (spreadsheet, tabular, grafix, and install) are written in Pascal and use the Techno-Jock version 5.0 toolkit. Therefore, the Borland Pascal version 7.0 compiler is used for these components of AGNPS. Each of these sub-programs are compiled separately in separate directories. The needed project files and desktop files must be kept along with the source code directories. Note that all compilation must be done locally (non-network) due to memory manager conflicts between the LAN drivers and the Borland drivers. To compile either the C or the Pascal source code, copy the program source directory from the network, invoke the compiler, change the compiler directory options to coincide with your local directory structure, and compile the program. Since the Borland C and the Borland Pascal products are used, the procedure is the same for all sub-programs.

Once compiling is complete, the model will need to be bound to the Phar Lap DOS Extender. A batch file is located along with the main model source code to do this. To bind along with Phar Lab simply type “phar”.

### DISCUSSION

The source code is located on the LAN in the H:\AGNPS\AGNPS500\SOURCE directory. All files are located in this directory as they were compiled and saved for the May release of the AGNPS version 5.00 program. A simple example of the H:\ directory structure is as follows;

```
H:\ AGNPS \ AGNPS500 \ SOURCE \ SCIENCE  - All AGRUN500 model source code
                              \ TABULAR  - Tabular output interface code
                              \ GRAFIX   - Graphical output interface src code
                              \ SPREAD   - Spreadsheet input interface code
                              \ SRCACCNT - Source accounting utility src code
                              \ INSTALL  - Installation utility source code
                              \ CHECK    - Checking utility source code
```

To compile a UNIX version of the AGRUN 5.0 model, the source code must be ported to the Sun machine using FTP. Note that all routines within the AGRUN (main model) source code are ANSI compatible between DOS and UNIX K & R. Therefore, to compile using the Standard C compiler available on the Solaris 4.1.3c operating system, no changes are required to the code. Use the following commands to compile a UNIX version of the AGRUN 5.00 model:

```
cc -D_UNIX_K_AND_R: *.c
cc *.o -lm
```

The first command will compile the C files of the model and the second command will link these files with the math libraries. The output executable file will be named “a.out” and may be changed to any desired filename. The file name chosen for this verification is AGRUN500.OUT.

## UNIX VERSION 5.00

AGRUN version 5.00 has been ported to the UNIX environment using the Solaris 4.1.3c operating system. The model was successfully compiled using the UNIX ANSI-Standard compiler on a Sun-LX Workstation. The DOS version source code has been made portable so that only one set of code will be maintained for DOS or UNIX. The additions needed for the UNIX compiled version 5.00 were mostly “#ifdef _UNIX_K_AND_R” syntax additions. The exception to this is in the DEBUGFLG.C file where the FSEEK function was used. The UNIX environment variables do not include the SEEK_SET constant. Therefore, an “#ifdef _UNIX_K_AND_R” case was added to define this value to 0.

### Discussion

The UNIX version of the AGNPS 5.00 model was tested and compared to the DOS version output files for the AGW7 and BEAVER data sets. This testing and comparison showed identical outputs for the UNIX and DOS versions of the AGNPS version 5.00 model. These output results will be placed on the ftp site along with these two data sets as agw7.output and beaver.output.

## PACKING AND RELEASE OF VERSION 5.00

The files to be released have been placed in two separate locations: (a) an FTP directory accessible by internet; (b) official files on the NCSCRL local area network (LAN). Only the FTP files are available to the public. The FTP files for users to download are on the UNIX AGNPS FTP directory and the official version for DOS is located on the LAN H:\ drive. Here is a directory tree structure of the locations of the files:

**UNIX FTP:**

```
/pub / agnps500 / dos          - FTP Self-extracting DOS distribution files
                / unix         - UNIX version FTP directory
                      / source - AGRUN model source code for compilation
                      / exe    - Sun-LX compiled version of AGRUN model
                / doc          - FTP Self-extracting DOS documentation files
```

**H:\(LAN)**

```
H:\ AGNPS \AGNPS500 \DISK1              - Distribution disk #1 mailing dir
                    \DISK2              - Distribution disk #2 mailing dir
                    \SOURCE             -- (as noted above in the compile methods for ver 5.0)
                    \DOCS               - Official documentation directory
                         \ FRACTALS     - Watershed generation program docs
                         \ VERIFICATION - Official verification report docs
                         \ REPORTS      - Various official report docs
                         \ MANUALS      - Official 5.0 user's  manuals dir
                         \ OTHER        - Other pertinent documentation
```

## Discussion

These directories have been created to be used and maintained by the AGNPS Support team. They will be granted exclusive write access to these locations. From now on, when speaking of the "Official Version", it will be found in the above locations.

The final addition made to the release of AGNPS version 5.00 is the inclusion of the AGW7 and BEAVER watershed data set output files with the AGNPS distribution diskettes. By adding these files to the AGNPS data set, we have "official" output files (AGW7.OUT and BEAVER.OUT) which may be compared to the user's output files (AGW7.NPS and BEAVER.NPS*). These files must match if the AGNPS program is properly installed and configured.* Therefore, if any problems are encountered by the user, this comparison should be run to verify proper use of the AGNPS version 5.00 model.

## Appendix A:

Variable Glossary Table 1: Hydrology

VERIFICATION TABLE NO. 1: Hydrology

NOTE: In the variable column below, the "@" symbol is used to represent "a function of".

| GROUP | VARIABLE | UNITS | DEFINITION | VARIABLE | FILE |
|---|---|---|---|---|---|
| Watershed | drainage area | acres | Total drainage area at the outlet. | drainage_area_below | cellcalc.c |
| | total cell area | acres | Area of basic cell. | base_cell_area | readinpu.c |
| | feedlot area | % of cell area | Sum of all feedlot subareas within cell. | @sum(current_feed_ptr) | readinpu.c |
| | impoundment area | % of cell area | Tot drainage area to outlet of the impndmnt. | impndptr->drainage_area | readinp.c |
| | receiving cell | id | Cell which current cell drains into. | receiving_cell_number | readinpu.c |
| | next routing cell | id | Cell which routes into current cell. | longest_path->cell_num | recroute.c |
| Overland | length | feet | Overland flow length. | sheet_flow_length | chan_tr55.c |
| | travel time | hours | Overland flow travel time. | time_overland | chan_tr55.c |

---

```
 Shallow Conc     length                   feet                Shallow concentrated flow length.
      shallow_length          chan_tr55.c

                  velocity                 fps                 Shallow concentrated flow velocity.
      velocity_shallow        chan_tr55.c

                  travel time              hours               Shallow concentrated travel time.
      time_shallow            chan_tr55.c
```

---

```
 Channel          top width                feet                Channel flow top width.
      width                   chan_tr55.c

                  depth at bankfull        feet                Channel flow depth at bankfull.
      depth                   chan_tr55.c

                  segment length           feet                Channel flow segment length.
      channel_length          chan_tr55.c

                  velocity                 fps                 Channel flow velocity.
      channel_velocity        chan_tr55.c

                  travel time              hours               Channel flow travel time.
      time_concentrated       chan_tr55.c

                  time of concentration    hours               Channel flow time of concentration.
      total_time              chan_tr55.c
```

---

```
 Hydrograph       peak discharge outlet    cfs                 Avg pk discharge of inlet & outlt hydrgrphs.
      flow_rate_below          cell_calc.c
                                     Inches/hour

                  tot runoff vol outlet    acre-feet           Avg tot runoff vol of inlt & outlt hydrgrph.
      prepeak_RO_function     cell_calc.c
                                           inches

            avg peak discharge      cfs                 Avg pk dischrg of inlet and outlet hydrgrphs.
      @avg(flow_rate)               cell_calc.c
```

```
              average runoff volume   Inches            Avg runoff vol of inlet and outlet hydrographs.
     @avg(runoff_volume_above)cell_calc.c
              runoff peak volume      % of runoff       Avg runoff pk vol of inlt & outlet hydrgrphs.
     prepeak_RO_fraction     cell_calc.c
              time to peak            hours             Avg time to pk of inlet & outlet hydrographs.
     time_to_peak_ave        cell_calc.c
              time to base            hours             Avg time to base of inlet & outlet hydrographs.
     duration                cell_calc.c
-------------------------------------------------------------------------------------------------------------
------------------------------------------
```

## Appendix B:

Variable Glossary Table 2: Sediment--all sources

VERFICATION TABLE NO. 2: Sediment--all sources

---

| PARTICLE SIZE CLASS | VARIABLE | UNITS | DEFINITION | VARIABLE | FILE |
|---|---|---|---|---|---|
| Clay | upstream clay amount | tons | Clay sediment amount coming into cell.. | upstream_yld[1] | run_out.c |
| | upstream clay concen | lbs-sed/lbs-H2O | Clay sediment concentration into cell. | upstream_yld[1]/runoff_flow_above | " |
| | tot in-cell clay sources | tons | Total clay sources generated within cell. | sed_gen_in[1] | run_out.c |
| | downstream clay amount | tons | Clay sediment amount leaving cell. | tot_sed_yield[1] | run_out.c |
| | downstream clay concen | lbs-sed/lbs-H2O | Clay sediment concentration leaving cell. | tot_sed_yield[1]/runoff_flow_below | " |
| Silt | upstream silt amount | tons | Silt sediment amount coming into cell. | upstream_yld[2] | run_out.c |
| | upstream silt concen | lbs | Silt sediment concentration into cell. | upstream_yld[2]/runoff_flow_above | " |
| | tot in-cell silt sources | tons | Total silt sources generated within cell. | sed_gen_in[2] | run_out.c |
| | downstream silt amount | tons | Silt sediment amount leaving cell. | tot_sed_yield[2] | run_out.c |
| | downstream silt concen | lbs-sed/lbs-H2O | Silt sediment concentration leaving cell. | tot_sed_yield[2]/runoff_flow_below | " |
| Small | upstream s-agg amount | tons | Sm. Agg sediment amount coming into cell. | upstream_yld[3] | run_out.c |
| Aggregate | upstream s-agg concen | lbs-sed/lbs-H2O | Sm. Agg sed concen. coming into cell. | upstream_yld[3]/runoff_flow_above | " |

```
            tot in-cell s-agg srces  tons                 Total sm. agg sources generated within.
     sed_gen_in[3]            run_out.c

            downstream s-agg amount  tons                 Sm. Agg sediment amount leaving cell.
     tot_sed_yield[3]         run_out.c

            downstream s-agg concen  lbs-sed/lbs-H2O            Sm. Agg sediment concen. leaving cell.
     tot_sed_yield[3]/runoff_flow_below "
---------------------------------------------------------------------------------------------------------
-----------------------------------------

 Large      upstream l-agg amount    tons                 Lg. Agg sediment amount coming into cell.
     upstream_yld[4]          run_out.c

 Aggregate  upstream l-agg concen    lbs-sed/lbs-H2O            Lg. Agg sed concen. coming into cell.
     upstream_yld[4]/runoff_flow_above      "

            tot in-cell l-agg srces  tons                 Total lg. agg sources generated within.
     sed_gen_in[4]            run_out.c

            downstream l-agg amount  tons                 Lg. Agg sediment amount leaving cell.
     tot_sed_yield[4]         run_out.c

            downstream l-agg concen  lbs-sed/lbs-H2O            Lg. Agg sediment concen. leaving cell.
     tot_sed_yield[4]/runoff_flow_below "
---------------------------------------------------------------------------------------------------------
-----------------------------------------

 Sand       upstream sand amount     tons                 Sand sediment amount coming into cell.
     upstream_yld[5]          run_out.c

            upstream sand concen     lbs-sed/lbs-H2O            Sand sediment concentration into cell.
     upstream_yld[5]/runoff_flow_above      "

            tot in-cell sand sources tons                 Total sand sources generated within cell.
     sed_gen_in[5]            run_out.c

            downstream sand amount   tons                 Sand sediment amount leaving cell.
     tot_sed_yield[5]         run_out.c

            downstream sand concen   lbs-sed/lbs-H2O.           Sand sediment concentration leaving cell.
     tot_sed_yield[5]/runoff_flow_below "
---------------------------------------------------------------------------------------------------------
-----------------------------------------

 Total      upstream total amount    tons                 Total sediment amount coming into cell.
     upstream_yld[6]          run_out.c

 Sediment   upstream total concen    lbs-sed/lbs-H2O            Total sediment concentration into cell.
     upstream_yld[6]/runoff_flow_above      "
```

tot in-cell total srces tons Total sediment sources generated within cell.
sed_gen_in[6] run_out.c

downstream total amount tons Total sediment amount leaving cell.
tot_sed_yield[6] run_out.c

downstream total concen lbs-sed/lbs-H2O Total sediment concentrn leaving cell.
tot_sed_yield[6]/runoff_flow_below "

------------------------------------------------------------------------------------------------------------------------------------------------

VERIFICATION TABLE NO. 2a: Sediment Transport--clay

------------------------------------------------------------------------------------------------------------------------------------------------

PARTITION

NUMBER VARIABLE UNITS DEFINITION VARIABLE
FILE

------------------------------------------------------------------------------------------------------------------------------------------------

```
 All         constant of proportionality     non-dim        Clay sediment concentration ratio.
     constant_prop[1]         newsoil.c

             particle fall velocity          ft/sec         Velocity clay falls from still water.
     set_velocity[1]         newsoil.c

             stream segment length           feet           Length of stream segment.
     flow_length             newsoil.c

             transport capacity factor       non-dim        Clay transport capacity factor.
     trans_cap[1]             newsoil.c

--------------------------------------------------------------------------------------------------------------------------------------------------
 Partition   unit-wdth H2O dischrg           cfs/ft         Clay unit width water discharge.
     downstream_qp/channel_width_out        "

  n (1-8)    unit-wdth trans capacity        lbs/sec/ft           Clay unit width transport capacity.
trans_capacity_out[1]    newsoil.c

             upstr unit-wdth dischrg         lbs/sec/ft           Clay upstream unit-wdth discharge.
     sed_flow_rate_in         newsoil.c

             dwnstr unit-wdth dischrg        lbs/sec/ft           Clay downstream unit-width discharge.
     sed_flow_rate_out       newsoil.c

             deposition number               non-dim        Time taken for clay particle to fall through
Nd                 newsoil.c
                                                               water column in time taken to travel through
                                                               cell <see SCS AGNPS Evaluation (rev. 4) pg. 12>.
--[repeat for each partition]--------------------------------------------------------------------------------------------------------------------
```

VERIFICATION TABLE NO. 2b: Sediment Transport--silt

| PARTITION NUMBER | VARIABLE | UNITS | DEFINITION | VARIABLE | FILE |
|---|---|---|---|---|---|
| All | constant of proportionality | non-dim | Silt sediment concentration ratio. | constant_prop[2] | newsoil.c |
| | particle fall velocity | ft/sec | Velocity silt falls from still water. | set_velocity[2] | newsoil.c |
| | stream segment length | feet | Length of stream segment. | flow_length | newsoil.c |
| | transport capacity factor | non-dim | Silt transport capacity factor. | trans_cap[2] | newsoil.c |
| Partition | unit-wdth H2O dischrg | cfs/ft | Silt unit width water discharge. | downstream_qp/channel_width_out | " |
| *n* (1-8) | unit-wdth trans capacity | lbs/sec/ft | Silt unit width transport capacity. | trans_capacity_out[2] | newsoil.c |
| | upstr unit-wdth dischrg | lbs/sec/ft | Silt upstream unit-wdth discharge. | sed_flow_rate_in | newsoil.c |
| | dwnstr unit-wdth dischrg | lbs/sec/ft | Silt downstream unit-width discharge. | sed_flow_rate_out | newsoil.c |
| | deposition number | non-dim | Time taken for silt particle to fall through water column in time taken to travel through cell <see SCS AGNPS Evaluation (rev. 4) pg. 12>. | Nd | newsoil.c |

--[repeat for each partition]--

VERIFICATION TABLE NO. 2c: Sediment Transport--sm. agg

| PARTITION NUMBER | VARIABLE | UNITS | DEFINITION | VARIABLE | FILE |
|---|---|---|---|---|---|
| All | constant of proportionality | non-dim | Sm. Agg sediment concentration ratio. | constant_prop[3] | newsoil.c |
| | particle fall velocity | ft/sec | Velocity Sm. Agg falls from still water. | set_velocity[3] | newsoil.c |
| | stream segment length | feet | Length of stream segment. | flow_length | newsoil.c |
| | transport capacity factor | non-dim | Sm. Agg transport capacity factor. | trans_cap[3] | newsoil.c |
| Partition | unit-wdth H2O dischrg | cfs/ft | Sm. Agg unit width water discharge. | downstream_qp/channel_width_out | " |

*n* (1-8) unit-wdth trans capacity lbs/sec/ft Sm. Agg unit width transport capacity.
trans_capacity_out[3] newsoil.c

upstr unit-wdth dischrg lbs/sec/ft Sm. Agg upstream unit-wdth discharge.
sed_flow_rate_in newsoil.c

dwnstr unit-wdth dischrg lbs/sec/ft Sm. Agg downstream unit-width discharge.
sed_flow_rate_out newsoil.c

deposition number non-dim Time taken for sm. agg particle to fall through
Nd newsoil.c

water column in time taken to travel through

cell <see SCS AGNPS Evaluation (rev. 4) pg. 12>.

--[repeat for each partition]------------------------------------------------------------------------------------------------------------

VERIFICATION TABLE NO. 2d: Sediment Transport--Lg. Agg

-----------------------------------------------------------------------------------------------------------------------------------------------

PARTITION

NUMBER VARIABLE UNITS DEFINITION VARIABLE
FILE

-----------------------------------------------------------------------------------------------------------------------------------------------

```
 All         constant of proportionality     non-dim        Lg. Agg sediment concentration ratio.
      constant_prop[4]         newsoil.c

             particle fall velocity          ft/sec         Velocity Lg. Agg falls from still water.
      set_velocity[4]         newsoil.c

             stream segment length           feet           Length of stream segment.
      flow_length             newsoil.c

             transport capacity factor       non-dim        Lg. Agg transport capacity factor.
      trans_cap[4]             newsoil.c
--------------------------------------------------------------------------------------------------------------------------------------------
 Partition   unit-wdth H2O dischrg           cfs/ft         Lg. Agg unit width water discharge.
      downstream_qp/channel_width_out       "

  n (1-8)    unit-wdth trans capacity        lbs/sec/ft           Lg. Agg unit width transport capacity.
trans_capacity_out[4]     newsoil.c

             upstr unit-wdth dischrg         lbs/sec/ft           Lg. Agg upstream unit-wdth discharge.
      sed_flow_rate_in         newsoil.c

             dwnstr unit-wdth dischrg        lbs/sec/ft           Lg. Agg downstream unit-width discharge.
      sed_flow_rate_out       newsoil.c

             deposition number               non-dim        Time taken for Lg. Agg particle to fall through
Nd                 newsoil.c
                                                water column in time taken to travel through
                                                cell <see SCS AGNPS Evaluation (rev. 4) pg. 12>.
--[repeat for each partition]--------------------------------------------------------------------------------------------------------------
```

VERIFICATION TABLE NO. 2e: Sediment Transport--sand

| PARTITION NUMBER | VARIABLE | UNITS | DEFINITION | VARIABLE | FILE |
| --- | --- | --- | --- | --- | --- |
| All | constant of proportionality | non-dim | Sand sediment concentration ratio. | constant_prop[5] | newsoil.c |
| | particle fall velocity | ft/sec | Velocity sand falls from still water. | set_velocity[5] | newsoil.c |
| | stream segment length | feet | Length of stream segment. | flow_length | newsoil.c |
| | transport capacity factor | non-dim | Sand transport capacity factor. | trans_cap[5] | newsoil.c |
| Partition | unit-wdth H2O dischrg | cfs/ft | Sand unit width water discharge. | downstream_qp/channel_width_out | " |
| *n* (1-8) | unit-wdth trans capacity | lbs/sec/ft | Sand unit width transport capacity. | trans_capacity_out[5] | newsoil.c |
| | upstr unit-wdth dischrg | lbs/sec/ft | Sand upstream unit-wdth discharge. | sed_flow_rate_in | newsoil.c |
| | dwnstr unit-wdth dischrg | lbs/sec/ft | Sand downstream unit-width discharge. | sed_flow_rate_out | newsoil.c |
| | deposition number | non-dim | Time taken for sand particle to fall through water column in time taken to travel through cell <see SCS AGNPS Evaluation (rev. 4) pg. 12>. | Nd | newsoil.c |

```
--[repeat for each partition]---------------------------------------------------------------------------------------------------------------
```

## Appendix C:

Variable Glossary Table 3: Chemicals

```
VERIFICATION TABLE NO. 3: Chemicals
---------------------------------------------------------------------------------------------------------------------------------------------

 CHEMICAL   GRP          VAR         UNITS               DEFINITION                                        VARIABLE
            FILE
---------------------------------------------------------------------------------------------------------------------------------------------
 Nitrogen   Dissolved   decay factor     %             Decay factor of soluble Nitrogen.
      channel->per_n_decay          decay.c
                        in from cells    lbs           Soluble Nitrogen flowing into cell.
      soluble_nitrogen_yield        recroute.c
                        generated local  lbs           Soluble Nitrogen generated within cell.
      soluble_nitrogen_runoff*area   run_out.c
                        out of cell      lbs           Soluble Nitrogen leaving cell.
      soluble_nitrogen_yield         run_out.c
```

```
        Attached    in from cells    lbs          Sediment attached Nitrogen flowing into cell.
      total_n_cell_outlet           recroute.c

                    generated local  lbs          Sediment attached Nitrogen generated within cell.
      total_n_within_cell             run_out.c

                    out of cell      lbs          Sediment attached Nitrogen leaving cell.
      total_n_cell_outlet           run_out.c

        Total       in from cells    lbs          Sum of above; total Nitrogen flowing into cell.
      N_dis_into + N_att_into       cellcalc.c

                    generated local  lbs          Sum of above; total Nitrogen generated within cell.
sum of above        run_out.c

                    out of cell      lbs          Sum of above; total Nitrogen leaving cell.
sum of above        run_out.c

                    enrichment ratio non-dim            Nitrogen enrichment ratio.
      nit_enrichment_ratio            sednut.c
--------------------------------------------------------------------------------------------------------------------------------------------------

 Phosphorus Dissolved   decay factor     %            Decay factor of soluble Phosphorus.       channel-
>per_p_decay        decay.c

                    in from cells    lbs          Soluble Phosphorous flowing into cell.
      soluble_phosphorus_yield            recroute.c

                    generated local  lbs          Soluble Phosphorus generated within cell.
      soluble_phosphorous_runoff*area    "

                    out of cell      lbs          Soluble Phosphorus leaving cell.
      soluble_phosphorous_yield      run_out.c

        Attached    in from cells    lbs          Sediment attached Phosphorus flowing into cell.
      total_p_cell_outlet           recroute.c

                    generated local  lbs          Sed attached Phosphorus generated within cell.
      total_p_within_cell             run_out.c

                    out of cell      lbs          Sediment attached Phosphorus leaving cell.
      total_p_cell_outlet           run_out.c

        Total       in from cells    lbs          Sum of above; total Phosphorus flowing into cell.
      P_dis_into + P_att_into       cellcalc.c

                    generated local  lbs          Sum of above; total Phosphorus generated within cell.
sum of above        run_out.c

                    out of cell      lbs          Sum of above; total Phosphorus leaving cell.
sum of above        run_out.c
```

enrichment ratio non-dim Sum of above; total Phosphorus leaving cell.
phos_enrichment_ratio sednut.c

---

Pesticides Dissolved decay factor % Decay factor of soluble Pesticides. [ Not used in 5.00] ***

in from cells lbs Soluble Pesticides flowing into cell.
soluble_pest recroute.c

generated local lbs Soluble Pesticides generated within cell.
soluble_pest run_out.c

out of cell lbs Soluble Pesticides leaving cell.
soluble_pest_exit run_out.c

Attached in from cells lbs Sediment attached Pesticides flowing into cell.
sediment_pest_exit recroute.c

generated local lbs Sed attached Pesticides generated within cell.
sediment_pest run_out.c

out of cell lbs Sediment attached Pesticides leaving cell.
sediment_pest_exit run_out.c

Total in from cells lbs Sum of above; total Pesticides flowing into cell.
Pest_dis_into + Pest_att_into cellcalc.c

generated local lbs Sum of above; total Pesticides generated within cell.
sum of above run_out.c

out of cell lbs Sum of above; total Pesticides leaving cell.
sum of above run_out.c

enrichment ratio non-dim Sum of above; total Pesticides leaving cell.
enrich_ratio pesticid.c

---

COD decay factor % Decay factor of Chemical Oxygen Demand.
channel->per_cod_decay decay.c

in from cells ppm Chemical Oxygen Demand flowing into cell.
soluble_cod_yield recroute.c

lbs [
Converted value ] recroute.c

generated local ppm Chemical Oxygen Demand generated within cell. [
Converted from below ] run_out.c

```
                                        lbs                                               cod_runoff  *  area
run_out.c    run_out.c
                        out of cell     ppm         Chemical Oxygen Demand leaving cell.
      tot_cod_ppm                   run_out.c
                                        lbs                                               soluble_cod_yield
run_out.c
--------------------------------------------------------------------------------------------------------------
-------------------------------------------
```