Utility Codes:
(Version 2002-1)

by
Dermott E. Cullen
University of California
Lawrence Livermore National Laboratory
L-159
P.O. Box 808
Livermore, CA 94550

Tele: 925-423-7359
E.Mail: cullen1@llnl.gov
Website: http://www.llnl.gov/cullen1

Introduction

TART is distributed with a number of utility codes that are designed to edit TART output for criticality or source problems. The documentation here completely supercedes the original documentation for these utility codes, which was published as Chapter 8: Utility Codes, of the TART95 documentation.

Changes in TART2002 Output Format

In order to accommodate many more zones the TART output format has been changed; earlier only three digit zone numbers were accommodated by the format; this has been extended to five digits.

As a consequence all of the TART utility codes distributed with TART2002 CD have been updated to recognize the new output format. **WARNING** - do not use older versions of the TART utility codes; they have been completely superseded.

PLOTTAB

From this documentation you should note that these utility codes make extensive use of the **plottab** code to present results in graphical form. You can make your job easier, and you can improve your understanding of results, if you learn to use **plottab**. All of the graphical results presented here where created by **plottab**.

Multiprocessing

MULTIPRO

Prepare input to start any number of TART runs. After ALL runs have finished, TARTSUM can be used to average the results of any number of runs.

TARTSUM
Combine results of a number of TART runs. This can be used to improve the statistical accuracy of your results by adding additional results. When using multiprocessing this can quickly improve results.

WARNING - Currently MULTIPRO and TARTSUM can only be used for source, not criticality, problems.

Criticality Problems

CRITEDIT
Summarize criticality results in tabular form.

BALANCE
Edit criticality production, absorption and leakage spectra for an entire system to PLOTTAB input format. Use PLOTTAB to see your results.

PATHC
Edit pathlength (flux) spectra for individual spatial zones; PATHC = Path for Criticality Problems. Use PLOTTAB to see your results.

Source Problems

FLUXEDIT
Edit flux and energy deposition for each spatial zone to PLOTTAB input format. Use TARTCHEK to overlay the results on your geometry.

PATHS
Edit pathlength (flux) spectra for individual spatial zones. Use PLOTTAB to see your results; PATHS = Path for Source Problems; neutrons and/or photon problems.

EDIT1112
Edit binary output to tabular form.

PLOT1112
Edit binary output to PLOTTAB input format. Use PLOTTAB to see your results.

Input Parameters

If there are input parameters, they will be in a file with the same name as the utility codes, plus an extension of .INP, e.g., for the BALANCE code, BALANCE.INP. Each input file contains a complete, up-to-date description of all input parameters. As such, input parameters will not be described here.

Input TART Results

TART.OUT is read by CRITEDIT, BALANCE, PATHC, PATHS, and FLUXEDIT. This is the TART default file name for TART output.

TART.OUT is written by TARTSUM. It reads a number of TART output files, and creates TART.OUT, containing the combined results.

EDIT1112 and PLOT1112 read binary output files created using TART tally type 11 or 12. These are TART options to score particles entering a zone and output their coordinates to a binary file for later use. Now tally type 11 or 12 can alternately be used to tally absorptions within zones, rather than particles entering a zone, allowing for improved simulation of detectors.

Multiprocessing

Unlike the multiprocessing methods supported by other codes, TART uses an approach that is independent of the type(s) of computer used, allows each computer processor to execute TART at maximum speed with no communication between processors. It even allows different types of computers to be used, and all of the results need not be produced at the same time. For example, if after running a number of problems you find you need more statistics, with TART's approach you can make additional runs and add these new results to your earlier results. Note, that since TART's approach is computer independent you need not worry that this approach will become outdated, as occurs with the approaches used by other codes.

TART's approach is to allow you to make any number of independent TART runs and to average the results together only after all runs have been completed. The averaging is done based solely upon the contents of the output files from each TART run. With this approach you can use as many processors as you can find; they need not even be the same type of processor. You can even repeatedly use the same processor(s) to add statistics. All you need to keep from each TART run is the TART output file; as long as you have these you can continue adding statistics to your results.

WARNING - Currently MULTIPRO and TARTSUM can only be used for source, not criticality, problems. For criticality problems you can use the utility code **critedit** to average together the results of a number of criticality problems.

MULTIPRO

MULTIPRO prepares input to be used to start any number of TART source problems, and to later combine the results using TARTSUM. This can be used to improve the statistical accuracy of your results by adding additional results. When using multiprocessing this can quickly improve results.

Input Parameters: **MULTIPRO.INP.** Output is only to your screen. Below is an example **MULTIPRO.INP** file,

```
   20    1tart02-4
=====(ONLY THE ABOVE LINE IS READ AS INPUT)====================
12345678901234567890 = COLUMN #S FOR FIRST LINE INPUT
```

```
COLUMNS   FORMAT   DEFINITION
==============================
  1 -  5    I5     # OF TART RUNS
  6 - 10    I5     STARTING RANDOM NUMBER SEQUENCE
 11 - 20    A10    NAME OF TART EXECUTABLE, E.G., tart02-4

  CREATE INPUT AND OUTPUT FOR MULTIPLE TART RUNS
  ALL USING THE SAME INPUT, EXCEPT FOR THE RANDOM # SEQUENCE
  (sentl 12) AND THE NAME OF TALLY TYPE 12 OUTPUT FILE (sentl51).

  THE BASIC TART INPUT DECK MUST BE NAMED TART.IN.
  THIS CODE WILL CREATE IN00000?, ? = 1, 2, 3, 4, ......
  EACH IDENTICAL TO TART.IN, BUT WITH A DIFFERENT sentl 12 and 51
  ADDED JUST BEFORE THE TART INPUT end LINE.

  IN ADDITION TO THE TART INPUT DECKS, THIS CODE WILL ALSO CREATE,
  TART.BAT   = A BATCH FILE TO EXECUTE THE SEQUENCE OF TART RUNS
  TARTSUM.INP  = TARTSUM INPUT TO AVERAGE ALL OF THE TART RUNS OUTPUT

  HERE'S THE SEQUENCE OF THINGS YOU SHOULD DO,
  1) EDIT MULTIPRO.INP TO DEFINE HOW MANY TART RUNS TO MAKE
  2) EXECUTE multipro TO CREATE ALL TART INPUT
  3) MAKE SURE YOUR TART EXECUTABLE IS NAMED ON MULTIPRO INPUT
  4) EXECUTE TART.BAT TO RUN THE SEQUENCE OF TART JOBS
  5) EXECUTE tartsum TO AVERAGE ALL OF THE OUTPUT FILES TOGETHER
  6) YOUR FINAL RESULTS WILL BE IN TART.OUT
  7) CLEAN UP BY DELETING THE IN00000? AND OUT0000? FILES.

  INPUT FILES
  ===========
  TART.IN      = TART INPUT PARAMETERS

  OUTPUT FILES
  ============
  TART.BAT     = BATCH FILE TO EXECUTE A # OF TART RUNS
  TARTSUM.INP  = TARTSUM INPUT DECK TO ADD ALL OUTPUTS TOGETHER

  IN00000?     = SAME AS TART.IN, WITH DIFFERENT RANDOM # SEQUENCE
                 ? = 1, 2, 3, ........

  WHEN TART.BAT IS EXECUTED, EACH TART RUN WILL CREATE AN OUTPUT
  FILE NAMED, OUT0000?, ? = 1, 2, 3, ........

  WHEN TARTSUM IS EXECUTED IT WILL USE THE CREATED INPUT DECK,
  TARTSUM.INP, TO AVERAGE ALL OF THE TART RESULTS TOGETHER.
```

MULTIPRO reads ONE TART input problem, named TART.IN, and produces ANY number of copies of the input deck where each copy differs from the original ONLY by the random number sequence used (sentl 12) and binary tally type name (sentl 51) and the name of the file, IN000001, IN000002, IN000003,….

MULTIPRO also produces a batch file named TART.BAT that can be used to start all of the input decks that it created. The contents of TART.BAT corresponding to the above input looks like this,

```
tart02-4   IN000001 OUT00001 > LST00001  &
tart02-4   IN000002 OUT00002 > LST00002  &
tart02-4   IN000003 OUT00003 > LST00003  &
tart02-4   IN000004 OUT00004 > LST00004  &
tart02-4   IN000005 OUT00005 > LST00005  &
tart02-4   IN000006 OUT00006 > LST00006  &
tart02-4   IN000007 OUT00007 > LST00007  &
tart02-4   IN000008 OUT00008 > LST00008  &
tart02-4   IN000009 OUT00009 > LST00009  &
tart02-4   IN000010 OUT00010 > LST00010  &
tart02-4   IN000011 OUT00011 > LST00011  &
tart02-4   IN000012 OUT00012 > LST00012  &
tart02-4   IN000013 OUT00013 > LST00013  &
```

```
tart02-4   IN000014 OUT00014 > LST00014  &
tart02-4   IN000015 OUT00015 > LST00015  &
tart02-4   IN000016 OUT00016 > LST00016  &
tart02-4   IN000017 OUT00017 > LST00017  &
tart02-4   IN000018 OUT00018 > LST00018  &
tart02-4   IN000019 OUT00019 > LST00019  &
tart02-4   IN000020 OUT00020 > LST00020  &
```

In this example,
1) The TART code, tart02-4 is used to simultaneously execute 20 problems.
2) IN000001 through IN000020 are 20 copies of the TART input (created by MULTIPRO)
3) OUT0001 through OUT00020 are 20 TART output listings (created by TART)
4) LST00001 through LST00020 are 20 dummy files for normal screen output (created by TART)

On UNIX and LINUX systems the & at the end of each line will run each problem in the background, thereby releasing your terminal to immediately continue on to start the next run, without delaying until the preceding run is completed. Essentially all 20 runs will be started at the same time.

On other systems, such as IBM-PC/Windows, currently & is ignored, and the problems are run sequentially one after another. Therefore on other systems the advantage of the simultaneous multiprocessing possible on UNIX and LINUX systems is currently not possible. However, note that the procedures described here can still be used to advantage by using any number of IBM-PC/Windows computers to simultaneously run problems for you, and to then use TARTSUM to add the results together.

MULTIPRO also produces an input file that can be used with TARTSUM to add all of the results together. For example, in the case described above where 20 problems are run, MULTIPRO will produce a file named TARTSUM.INP. This file will contain the filenames of the 20 TART output listings to add together. For the above example this file would contain,

```
OUT00001
OUT00002
OUT00003
OUT00004
OUT00005
OUT00006
OUT00007
OUT00008
OUT00009
OUT00010
OUT00011
OUT00012
OUT00013
OUT00014
OUT00015
OUT00016
OUT00017
OUT00018
OUT00019
OUT00020
```

To use MULTIPRO, edit **MULTIPRO.INP** to define how many problems you want to run, and which random number sequence to start with, and the name of the TART code to use, e.g., tart02-4; each successive problem will be assigned the next random number sequence in sequential order, and type,

multipro

On any UNIX or LINUX system when MULTIPRO finishes follow these steps,

1) Make sure TART.BAT is executable [chmod 777 TART.BAT] and execute it to start all of the problems running.
2) Use "top" to monitor execution of the problems.
3) When ALL of the problems have finished, execute TARTSUM to average all of your results together and produce one final TART output file named TART.OUT.

It is as simple as that. Used properly this can be an extremely simple and yet powerful tool that can greatly accelerate how much work you can accomplish in a short period of time. For example, if you have available one or more of the currently available computers that have thousands of processors, you could use all of the processors to accelerate your work by factors of thousands.

TARTSUM

TARTSUM will combine results of a number of TART runs. This can be used to improve the statistical accuracy of your results by adding additional results. When using multiprocessing this can quickly improve results.

The TARTSUM code reads a series of entire TART output files, for **any number of source problems**, and creates a combined file named TART.OUT. All of the output files MUST correspond to EXACTLY the same TART input problem(s) - the ONLY thing that you are allowed to change is the random number sequence (sentl 12) [to make the results statistically independent], and binary tally type name (sentl 51).

Input Parameters: **TARTSUM.INP**

TARTSUM.INP contains a list of the names of the TART output files you want to sum – see the above description of MULTIPRO. Here's a simple example of the contents of TARTSUM.INP, corresponding to the MULTIPRO input described above,

```
OUT00001
OUT00002
OUT00003
OUT00004
OUT00005
OUT00006
OUT00007
OUT00008
OUT00009
OUT00010
OUT00011
OUT00012
OUT00013
OUT00014
```

```
OUT00015
OUT00016
OUT00017
OUT00018
OUT00019
OUT00020
```

To use TARTSUM, edit **TARTSUM.INP** to define the output you want, be sure that all of the files you want it to read are in the same directory, and that NONE are named TART.OUT (the combined output file name), and type,

tartsum

You will obtain results in a file named TART.OUT, that is in EXACTLY the same format as any other TART output file. Therefore, if you have any codes that process TART output - not to worry - they will work on the combined file, just as they work on any other TART output files.

Criticality Problems

CRITEDIT

CRITEDIT reads an entire TART output file, TART.OUT, for **any number of criticality problems**, and produces a table summarizing the results of the calculations.

Input parameters: **NONE,** Output to screen and file **CRITEDIT.LST.**

To use CRITEDIT, be sure that the file you want it to read is named TART.OUT, and type,

critedit

You will obtain results both on your screen as the code runs, and in an output file named CRITEDIT.LST (CRITEDIT list).

There are two primary use for CRITEDIT, both are illustrated in the TART95 documentation,
1) Summarize the results of a variety of different criticality calculations.
2) Summarize the results of running the same criticality calculation a number of times. This can be used to further check on and improve the statistical accuracy of results

Below is an example output. In this case KENO's model of Jezebel has been repeated 20 times. This example illustrates the power of TART; in this case it took TART only 22.06 seconds to complete all 20 repetitions of this problem – about 1.1 second per calculation (this was on a 600 MHz PC; a more modern PC would run TART 5 times faster than this). Note the average of the 20 runs, K-eff = 0.9999668. Note also the excellent agreement of the results of the 20 runs compared to a normal distributions, which is what we expect.

Criticality Calculation Editor (CRITEDIT 2002-1)

Crit. ID.	Fuel	Reflector or Name	Expected K	Removal Lifetime (Microsec.)	Median Energy (MeV)	Average Energy (MeV)	Seconds
Jezebel	Pu239	KENO-Jezebel	1.0003500	3.99549E-03	1.27055E+00	1.77974E+00	1.810
Jezebel	Pu239	KENO-Jezebel	0.9982060	3.96826E-03	1.26342E+00	1.78239E+00	3.030
Jezebel	Pu239	KENO-Jezebel	0.9990330	3.99949E-03	1.26300E+00	1.77712E+00	4.160
Jezebel	Pu239	KENO-Jezebel	1.0044200	4.02828E-03	1.29169E+00	1.81499E+00	5.090
Jezebel	Pu239	KENO-Jezebel	0.9974600	3.95871E-03	1.27873E+00	1.79530E+00	5.890
Jezebel	Pu239	KENO-Jezebel	1.0007600	3.97011E-03	1.26349E+00	1.78649E+00	6.620
Jezebel	Pu239	KENO-Jezebel	0.9963940	4.00936E-03	1.25544E+00	1.76781E+00	7.220
Jezebel	Pu239	KENO-Jezebel	1.0024400	3.99778E-03	1.26822E+00	1.78409E+00	8.450
Jezebel	Pu239	KENO-Jezebel	1.0004300	4.00257E-03	1.26716E+00	1.78647E+00	9.220
Jezebel	Pu239	KENO-Jezebel	0.9998420	3.95423E-03	1.29067E+00	1.80477E+00	10.340
Jezebel	Pu239	KENO-Jezebel	1.0054500	4.00914E-03	1.28417E+00	1.79204E+00	11.700
Jezebel	Pu239	KENO-Jezebel	1.0012100	3.98780E-03	1.27976E+00	1.78539E+00	12.890
Jezebel	Pu239	KENO-Jezebel	0.9945260	3.97229E-03	1.25676E+00	1.77525E+00	14.410
Jezebel	Pu239	KENO-Jezebel	1.0042300	4.01928E-03	1.26431E+00	1.79214E+00	16.030
Jezebel	Pu239	KENO-Jezebel	1.0009700	4.02265E-03	1.25465E+00	1.78051E+00	16.690
Jezebel	Pu239	KENO-Jezebel	0.9979210	3.99950E-03	1.27594E+00	1.80878E+00	17.770
Jezebel	Pu239	KENO-Jezebel	0.9949250	3.97579E-03	1.27800E+00	1.79364E+00	19.160
Jezebel	Pu239	KENO-Jezebel	1.0029900	3.97212E-03	1.26912E+00	1.78525E+00	20.000
Jezebel	Pu239	KENO-Jezebel	0.9987130	3.99786E-03	1.26674E+00	1.78116E+00	21.060
Jezebel	Pu239	KENO-Jezebel	0.9990660	3.97190E-03	1.25610E+00	1.78467E+00	22.060
		Average	0.9999668	+/- 0.0023582		Total	22.060
		Lowest	0.9945260	-0.0054408 (from Average)			
		Highest	1.0054500	0.0054832 (from Average)			

```
 Frequency Distribution for Expected K Values from 0.9 to 1.1
 (0.005 bin width sums)

 Expected K Range    Occurrences
  0.990  0.995       2 XX
  0.995  1.000       8 XXXXXXXX
  1.000  1.005       9 XXXXXXXXX
  1.005  1.010       1 X
 ------------------------------------------------------------
           Sum      20 (inside  0.9 to 1.1 Range)
                     0 (outside 0.9 to 1.1 Range)
 ------------------------------------------------------------
 Average  1.0000 +/-  0.0007 s. d. (0.0001 bin width average)

 K(i)                = K-Effective for Sample i
 N                   = Number of Samples        =      20
 Average K   = <K>   = K(i)/N  , Sum i=1 to N   =  1.0000
 Average K^2 = <K^2> = K(i)^2/N, Sum i=1 to N   =  1.0000
 Standard Deviation  = Sqrt[(<K^2>-<K>^2)/(N-1)]=  0.0007
 Sample Variance     =     [(<K^2>-<K>^2)]      =  0.0000
 Sample Width        = Sqrt[(<K^2>-<K>^2)]      =  0.0029

 As N Approaches Infinity Standard Deviation Approaches Zero.
 As N Approaches Infinity Sample Width Approaches a Constant.

 Confidence Limits
 (Occurrences out to +/- 10 times Sample Width)

 Sample Width  Occurrences  Per-Cent   Per-Cent
 Range                      Occurred   Normal
  -2  -1            3         15.000     13.591
  -1   0            7         35.000     34.134
   0   1            6         30.000     34.134
   1   2            4         20.000     13.591
 ---------------------------------------------
    Sum            20
```

BALANCE

For any criticality calculation the multiplication of the system (K-eff), is defined as a BALANCE between the **production** of neutrons and loss from the system due to **absorption and leakage**. BALANCE will allow you to see the results for any system as a function of neutron energy. Results include a spectrum of the incident energy at which neutron interactions produced neutrons (not the fission spectrum produced), as well as spectra of the energy at which neutrons were absorbed or leaked from the system. By comparing these three curves it is very simple to see what energy ranges are important. For example, those energy ranges where production exceeds the sum of absorption and leakage will tend to make a system super-critical, and those energy ranges where production is less than the sum of absorption and leakage will tend to make it sub-critical. The integral over energy of these results defines the multiplication of the system (K-eff).

The BALANCE code reads an entire TART output file, TART.OUT, for **one criticality problem**, and produces output in the PLOTTAB input format, so that you can immediately see your results.

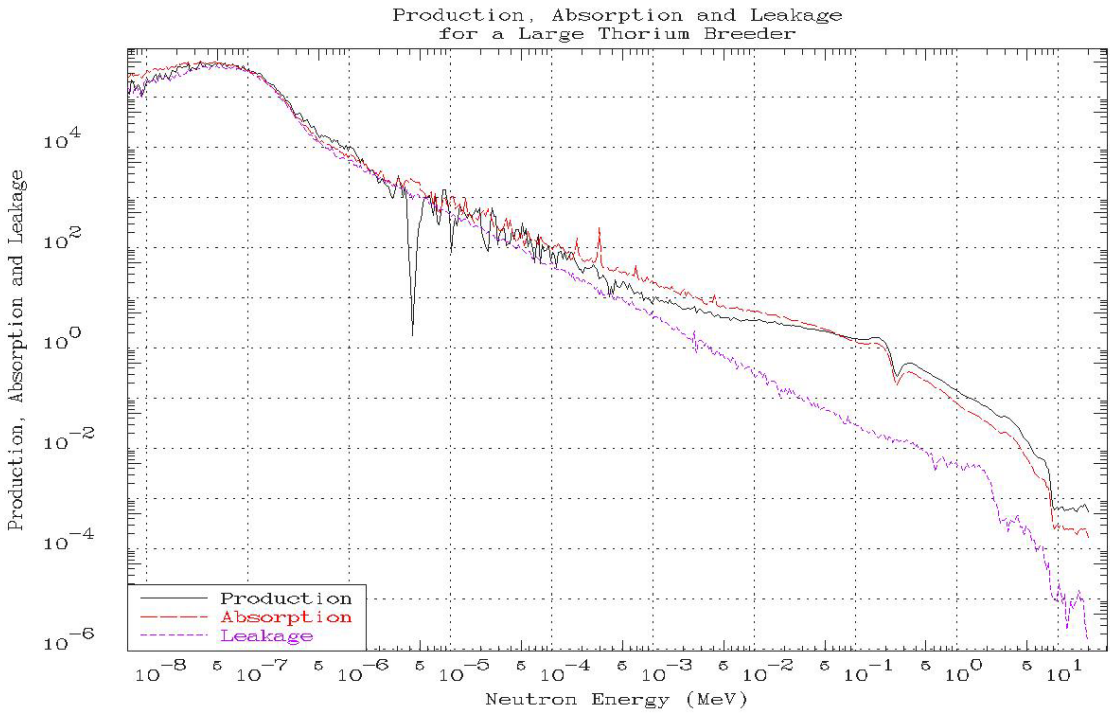
Input Parameters: **BALANCE.INP .** Output: Only to screen – below is an example of the input,

```
001110     Expected
000000     Analog
==========(ONLY THE ABOVE 2 LINES ARE READ)============================
The input controls what is output. 0 = No output, 1 = output.

Production, Absorption and Leakage results can be output
for Expectation and/or Analog calculations - the first input
line controls Expected output and the second line Analog.

Column
================================
  1       Total Cross Section
  2       Total Mean Free Path
  3       Neutron Production
  4       Neutron Absorption
  5       Neutron Leakage
  6       Normalize per MeV [0 = yes, 1 = no]

WARNING - Total Cross Section is ONLY for the first defined material.
```

To use BALANCE, edit **BALANCE.INP** to define the output you want, be sure that the file you want it to read is named TART.OUT, and type,

balance

You will obtain results in a file named PLOTTAB.CUR, which can be moved to your PLOTTAB directory and immediately plotted. Below is a plot of the results of a sample problem; this is the example problem included with plottab; see, the PLOTTAB documentation for details.



PATHC

BALANCE is designed to allow you to see overall features of an entire system. In contrast, PATHC allows you to see the features of individual spatial zones. It will edit pathlength (flux) spectra as a function of energy for individual spatial zones, and allow you to optionally renormalize the results per unit energy and/or unit volume.

Input Parameters: **PATHC.INP.** Output is only on the screen. Below is an example input,

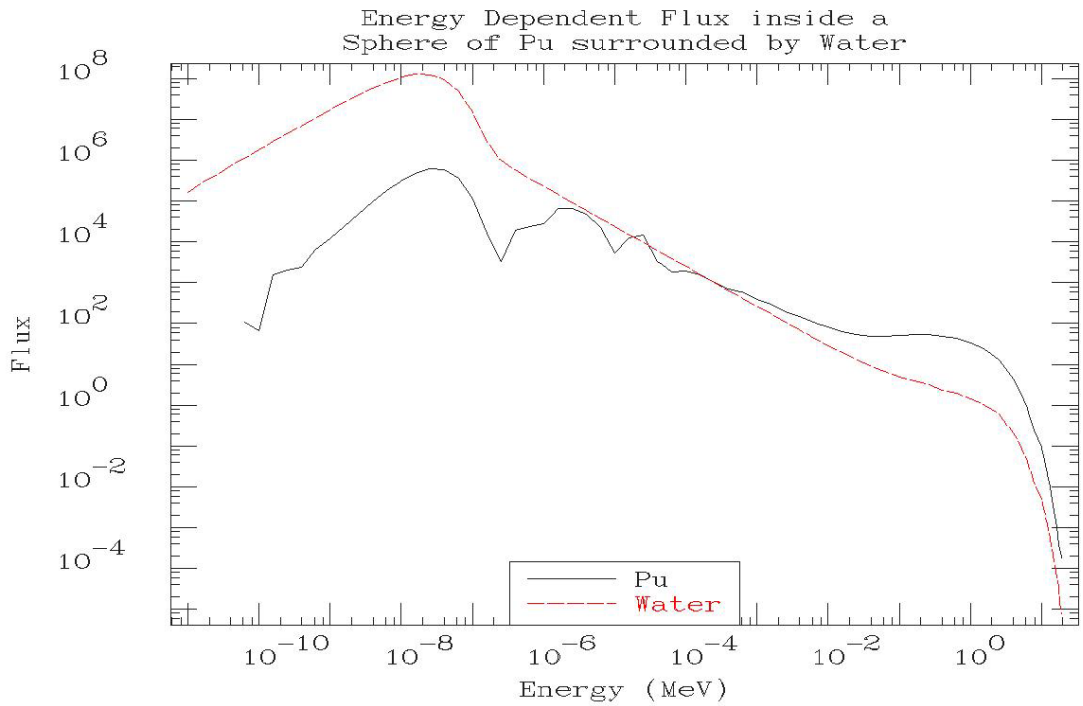
```
1  (0 = do not divide by width of group, 1 = divide)*dependent flux
1  (0 = do not divide by volume of zone, 1 = divide)*for any zones
1  (zones numbers, 1 per line, up to 1,000 allowed) *you select.
2
3
4
5
6
7
8
9
10
```

To use PATHC, edit **PATHC.INP** to define the output you want, be sure that the file you want it to read is named TART.OUT, and type,

pathc

You will obtain results in a file named PLOTTAB.CUR, which can be moved to your PLOTTAB directory and immediately plotted; see, the PLOTTAB documentation for details.

The below figure shows the flux normalized per unit energy and volume for a sphere of Pu surrounded by 10 cm of water. This plot shows the average flux in the Pu and the average flux in the water. In the water we can see a fully formed thermal Maxwellian. In contrast the average flux in the Pu is still quite fast. For a system similar to this one you can add additional spatial zones and obtain plots showing details of the flux versus radius in both Pu and water.



WARNING - the name of this utility code has been changed to PATHC from the earlier name PATH. The earlier name caused problems on some systems where attempts to run a code named path by typing,

path

resulted in changing the system defined “path”, which is NOT a good idea!!!

Source Problems

FLUXEDIT

FLUXEDIT is designed to edit flux and energy deposition for each spatial zone to PLOTTAB input format, which can be used by TARTCHEK to overlay the results on your geometry. This is a very powerful capability that you should not overlook.

The FLUXEDIT code reads an entire TART output file, TART.OUT, for **one source problem**, and produces output in the PLOTTAB input format, so that you can immediately see your results.

Input Parameters: **FLUXEDIT.INP**. Output is only on your screen. Below is an example input,

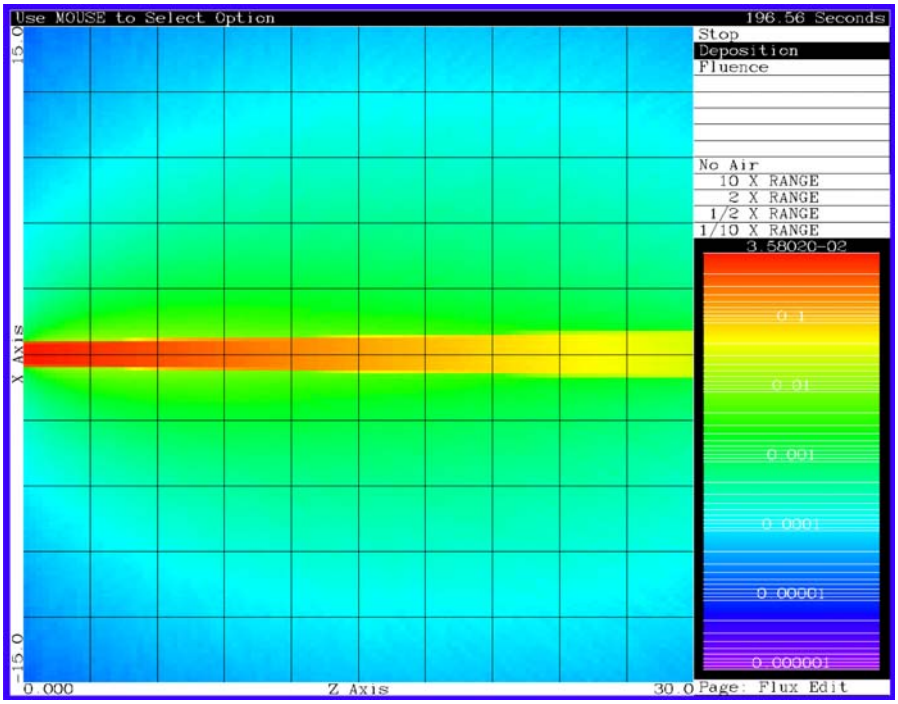
```
111
============(ONLY THE ABOVE LINE IS READ)=============
Definition of input parameters
 Column       Format    Definition
   1            I1      output flux (fluence)   (0=no, 1=yes)
   2            I1      output energy deposit   (0=no, 1=yes)
   3            I1      normalize output per cc (0=no, 1=1/cc, 2=1/gram)

                        Note, the new option to normalize per unit mass
```

To use FLUXEDIT, edit FLUXEDIT.INP to include the options you want to use, and be sure that the file you want it to read is named TART.OUT, and type,

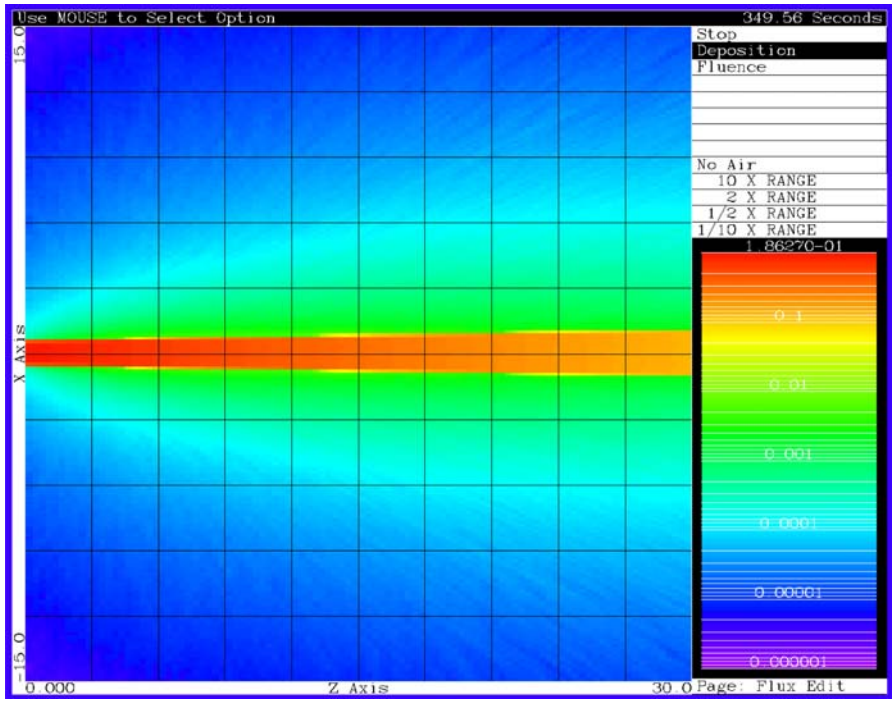
fluxedit

You will obtain results in a file named FLUXEDIT.OUT, which can be moved to your TARTCHEK directory and immediately used to overlay your results onto your geometry. See the chapter on **TARTCHEK** to learn how to overlay your results on your geometry.

The below two figures both used the same cylinder of water, 15 cm in radius and 30 cm in height. So that we can see the spatial variation in the flux the cylinder was divided into 10,000 spatial zones (100 x 100 in R x Z). In each case photons are incident from the left, on the axis of the cylinder, in a narrow beam directed along the axis of the cylinder. The two figures differ only in that the first figure shows results for 1 MeV photons incident and the second shows results for 10 MeV photons incident. Figures such as these can greatly enhance your understanding of what is happening inside any system. These two figures are from the TARTCHEK report; for additional, similar results see the TARTCHEK report.



1 MeV Photon Incident on Water



10 MeV Photon Incident on Water

PATHS

Similar to PATHC for **criticality** problems, PATHS allows you to see the features of individual spatial zones for **source** problems. It will edit pathlength (flux) spectra as a function of energy for individual spatial zones, and allow you to optionally renormalize the results per unit energy and/or unit volume. You can obtain output for neutron and/or photon **source** problems.

Input Parameters: **PATHS.INP.** Output is only on the screen. Below is an example input file,

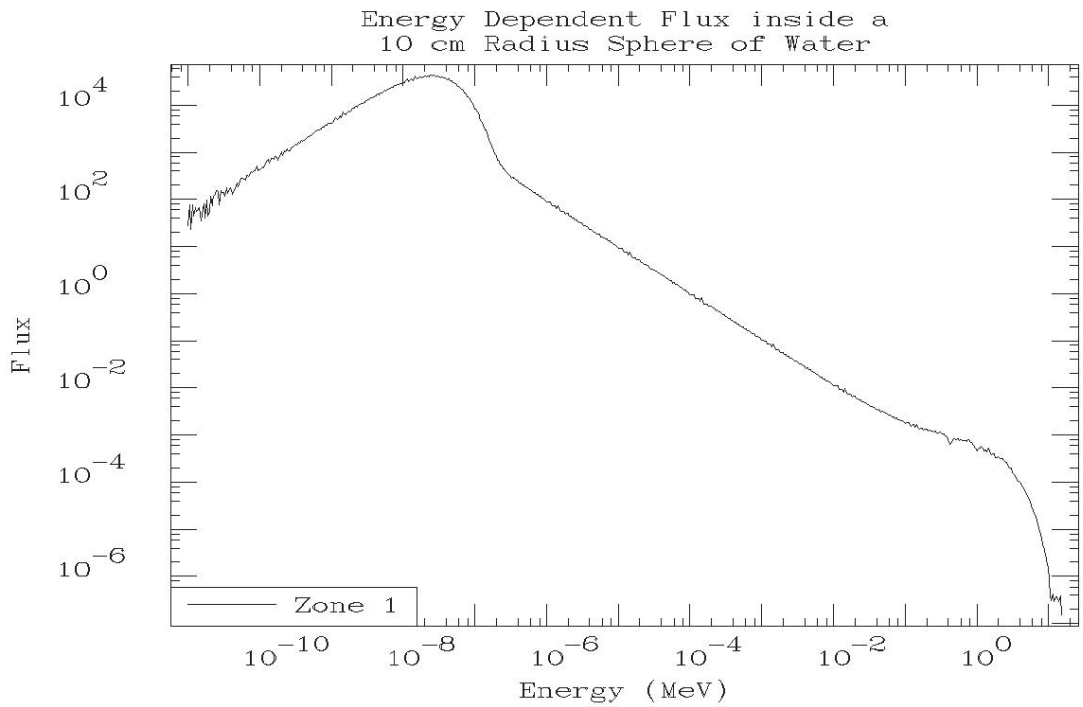
```
 0  (0 = neutron results, 1 = photon results)        *energy
 1  (0 = do not divide by width of group, 1 = divide)*dependent data
 1  (0 = do not divide by volume of zone, 1 = divide)*for any zones
 1  (zones numbers, 1 per line, up to 1,000 allowed) *you select.
 2
 3
 4
 5
 6
 7
 8
 9
10
```

To use PATHS, edit **PATHS.INP** to define the output you want, be sure that the file you want it to read is named TART.OUT, and type,

paths

You will obtain results in a file named PLOTTAB.CUR, which can be moved to your PLOTTAB directory and immediately plotted; see, the PLOTTAB documentation for details.

Below is a plot of the flux inside a 10 cm radius sphere of water due to the fission source at the origin. These results are for the same problem as the below plot1112 results, but here I have divided the results by the width of each tally bin, thereby converting the results from per unit lethargy (as shown for the below plot1112 results) to per unit energy here. Comparing the results for paths and plot1112 we see similar features; they differ primarily in the units of flux: per unit lethargy or energy. Note the difference of a factor of 1/E, e.g., here the 1/E slowing down spectrum is fairly obvious, which compares to the constant slowing down in the plot1112 output.



EDIT1112

TART tally types 11 and 12 for neutrons or photons allows users the option of tallying particles when they enter individual spatial zones; the particle is then terminated only if the zone is empty. For each particle the output includes all spatial, energy, time, direction, and the number of collisions from source. This option is used by many TART users for special applications, such as detector response problems. Starting with the 2000 distribution, tally type 11 or 12 can alternately be used to tally absorptions within zones, rather than particles entering a zone, allowing for improved simulation of detectors; see TART input parameters for details.

EDIT1112 was initially designed to read a binary file created by TART and produce tabulated results that users can then use in their applications. However, currently it is more often used by users within their special purpose application codes. EDIT1112 is incorporated into their codes and is used to read the coordinates for one particle at a time from a binary file, and the users then do whatever they want with them.

The EDIT1112 code reads **a single binary** TART output file, for **a single source problem**, and creates a file of tabulated results. The user defines by input the file name of the output tabulated result.

Input Parameters: **EDIT1112.INP.** Below is an example input file,

```
12g
gco12000
GAMMA12
====(NOTHING BELOW THIS LINE IS READ)==================================

     TART NEUTRON/PHOTON TALLY TYPE 11 AND 12 EDITOR
```

```
     INPUT PARAMETERS
     ================
     LINE    COLUMNS    FORMAT    DESCRIPTION
     ========================================
       1      1-3        A3       TYPE OF DATA
                                  11n = TALLY TYPE 11, NEUTRONS
                                  11g = TALLY TYPE 11, PHOTONS
                                  12n = TALLY TYPE 12, NEUTRONS
                                  12g = TALLY TYPE 12, PHOTONS
                                  OTHERWISE = ERROR
       2      1-8        A8       FILENAME OF BINARY TART96 OUTPUT
       3      1-8        A8       FILENAME OF OUTPUT REPORT

For example, the above input
12g          Tally type 12, photons
gco12000     Name of the binary file output by tart96
GAMMA12      Name of the listing file output by this code
```

To use EDIT1112, edit **EDIT1112.INP** to define the output you want, be sure the binary file you want it to read is in the same directory, and type,

edit1112

You will obtain tabulated results in the file you specified by input.

WARNING - EDIT1112 will only read **a single binary** file, even though TART may create an entire family of binary files. This is done because initial tests of the code illustrated that reading and tabulating the results for an entire family of files can result in enormous output files. To use an entire family of file, see PLOT1112 below.

PLOT1112

PLOT1112 is merely one example of a special purpose code built on top of EDIT1112. PLOT1112 reads **a family of binary** files created by TART, summarize the results in a file name defined by user input, and also produces results versus time, in a file named **TIME.CUR**, versus energy, in a file named **ENERGY.CUR**, and versus the direction cosine with respect to the Z axis, in a file names **ZCOSINE.CUR**. Each of these files is in the PLOTTAB format, so that you can use PLOTTAB to see your results. Each file contains PLOTTAB "curves" for total results, as well as results for each number of collisions that particles had between source and entering the tally zone, e.g., uncollided, first collided, second collided, etc., results.

Input Parameters: **PLOT1112.INP.** Below is an example input file,

```
12n    1000  45.0               0
ncord000
NEUT12
====(NOTHING BELOW THIS LINE IS READ)==================================

     TART NEUTRON/PHOTON TALLY TYPE 11 AND 12 EDITOR

     INPUT PARAMETERS
     ================
     LINE    COLUMNS    FORMAT    DESCRIPTION
     ========================================
       1      1-3        A3       TYPE OF DATA
                                  11n = TALLY TYPE 11, NEUTRONS
                                  11g = TALLY TYPE 11, PHOTONS
                                  12n = TALLY TYPE 12, NEUTRONS
```

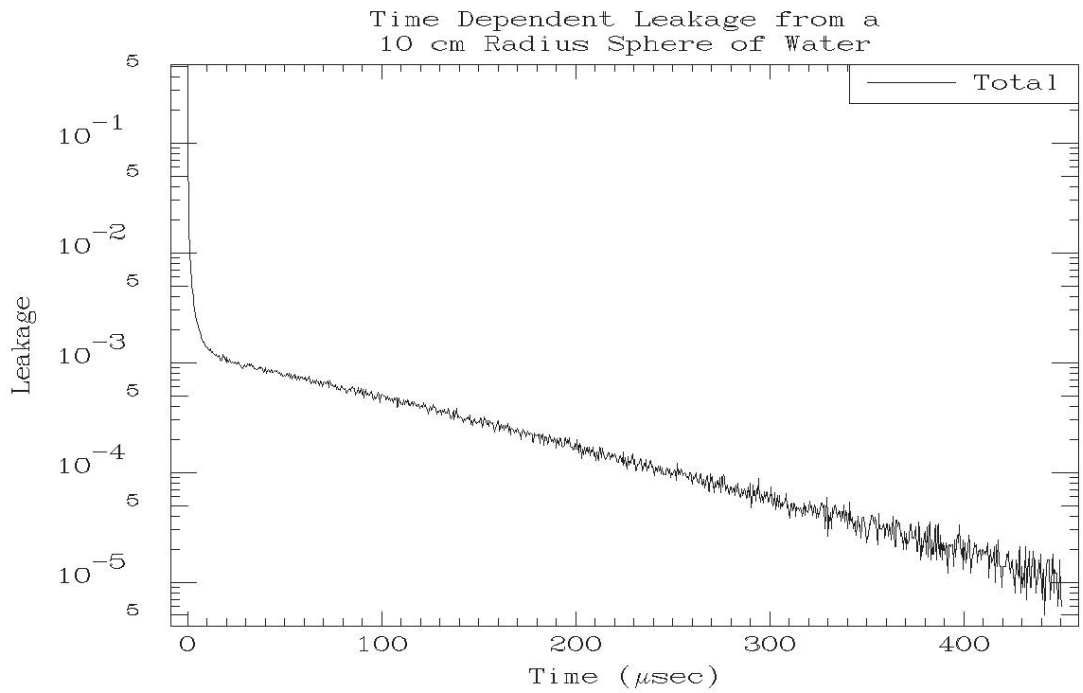
```
                                12g = TALLY TYPE 12, PHOTONS
                                OTHERWISE = ERROR
       1      4-11       I8      NUMBER OF BIN TO ACCUMULATE RESULTS
                                = 10 TO 1000
       1     12-22      D11.4    MAXIMUM TIME (IN SHAKES = 1.0D-8 SEC)
                                = 0 - RANGE OF DATA READ
                                > 0 - DEFINES RANGE OF OUTPUT
       1     23-33       I11     DUPLICATES
                                = 0 - USE
                                > 0 - IGNORE
       2      1-8        A8      FILENAME OF BINARY TART OUTPUT
       3      1-8        A8      FILENAME OF OUTPUT REPORT

For example, the above input
12n    1000  45.0
12n          Tally type 12, neutrons
1000         1000 Tally bins
45.0         45 shakes is maximum tally time
ncord000     Name of the binary file output by tart96
NEUT12       Name of the listing file output by this code
```

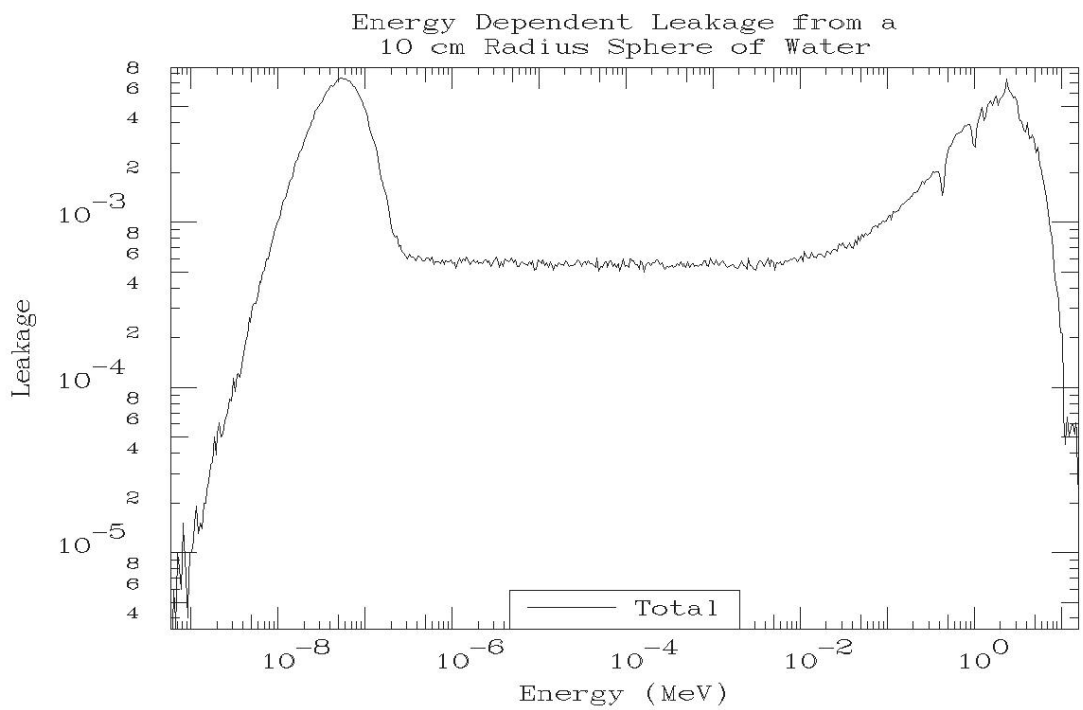
To use PLOT1112, edit **PLOT1112.INP** to define the output you want, be sure the entire family of binary files you want it to read is in the same directory, and type,

plot1112

You will obtain a summary of results in the file you specified by input, as well as **TIME.CUR** and **ENERGY.CUR**, that can be used with PLOTTAB to see your results. Note, **TIME.CUR** and **ENERGY.CUR** correspond to the standard PLOTTAB input file **PLOTTAB.CUR**. By moving either to your PLOTTAB directory and renaming it **PLOTTAB.CUR** you can immediately see your results. See, the PLOTTAB documentation for details.

Below is an example of time dependent output (from **TIME.CUR**), in this case the time dependent leakage from a 10 cm radius sphere of water starting from a fission source at the origin of the sphere. From this plot we can see the initial very rapid decrease, followed by the fundamental mode decay. The following plot is of the energy dependent leakage spectrum (from **ENERGY.CUR**). Note that the energy spectrum is per unit lethargy, which differs from the spectrum per unit energy only by a factor of 1/E (for results per unit energy, see the above **paths** results). Here we can see the high energy fission spectrum, followed by the slowing down spectrum (constant in lethargy, 1/E in energy), and finally the thermal Maxwellian. These and similar plots can help us learn a great deal about what is happening in our systems.


Time Dependent Leakage from a
10 cm Radius Sphere of Water
Total
Leakage
Time (μsec)


Energy Dependent Leakage from a
10 cm Radius Sphere of Water
Total
Leakage
Energy (MeV)