Introduction to PEGS4
SNO-STR-96-019

Mike Lay

June 6, 1996

SNOMAN relies on the EGS4 code package for its simulation of electrons, positrons and gammas. EGS4 in turn relies on a package called PEGS4, which is used to produce cross section data files. This document is designed to help in the use of PEGS4.

In principle, the user can find all he or she needs to know in the EGS4 manual [1]. In practise, the manual is a little obscure in places, and fails to mention a few of the niceties! However, the information in this document is drawn mostly from the author's experience and prejudice, and is therefore also not complete.

1 Setting Up the Input File

PEGS comes complete with a pair of data files (PGS4FORM.DAT and PGS4PEPR.DAT) containing most of the data PEGS needs. However, the user must prepare an input file specifying the details of the material[1] required. This input file consists of a 'card' for each material, which looks something like:

```
COMP
 &INP NE=2,RHO=1.0,PZ=2,1 &END
LIGHT-WATER
H  O
ENER
 &INP AE=0.521,UE=100000.,AP=0.01,UP=100000. &END
PWLF
 &INP NALE=100,NALG=100 &END
DECK
 &INP  &END
```

This card can be split up into four commands, COMP, ENER, PWLF and DECK, described below. There are additional commands referring to plotting functions, but these have no use in generating the data files. Note that PEGS is case sensitive; use upper case at all times.

1.1 COMP

This section describes the material. COMP stands for compound, and could be replaced with ELEM (element) or MIXT (mixture). The three are discussed separately as they have different functionalities.

[1] You can have more than one material in an input file, but this activates an obscure bug, discussed in section 3.

1.1.1 ELEM

The simplest option. PEGS has parameters on file for $Z = 1 \rightarrow 100$, and will automatically find densities, isotopic abundances, etc, though these can be supplied if required. A sample is shown below. The third line specifies the name to be given to the medium, the fourth line specifies the element.

```
ELEM
 &INP &END
ALUMINIUM
AL
```

1.1.2 COMP

The COMP option is used for a material which contains more than one type of atom, whose elemental composition is to be specified by the relative numbers of atoms. The required information is then NE (the number of elements), RHO (the density) and PZ (the relative proportions of the elements). The third line specifies the name, the fourth line specifies the elements involved. Note that each element occupies three spaces, even if its symbol is a single letter.

```
COMP
 &INP NE=2,RHO=1.0,PZ=2,1 &END
LIGHT-WATER                    H2O
H  O
```

1.1.3 MIXT

The MIXT option is similar to the COMP option except that the relative atomic proportions are given by mass (RHOZ(I)) rather than number. The third and fourth lines are the name and elements involved, as before.

```
MIXT
 &INP NE=4,RHO=7.9,RHOZ=68.0,19.0,1.7,10.5 &END
SSTEEL
FE CR MN NI
```

1.1.4 Gasses

For gasses, the pressure in atmospheres (GASP) must also be specified:

```
COMP
 &INP NE=3,RHO=1.29E-3 RHOZ=0.755,0.231,0.012 GASP=1.&END
AIR
N  O  A
```

1.2 ENER

This section is only required for the first material. It specifies the maximum energies for the simulation (UE and UP for electrons and photons respectively) and the cutoff energies (AE and AP for electrons and photons respectively). Discussion of the latter is a topic beyond the scope of this document. If this section is not required then it may be replaced with '&INP

&END'. All energies are total energies, specified in MeV. Optionally, a negative number may be given which is interpreted as a number of electron masses. Note that as EGS4 does not include electro-weak interactions, setting the upper energy limits much above 10 GeV will give incorrect results.

1.3 PWLF

This is the instruction to PEGS to perform a Piece Wise Linear Fit to the cross sectional data. NALE and NALG are optional parameters specifying the number of evaluation points on the energy scale for electrons and photons respectively. Note that the points are evenly distributed along a logarithmic energy scale from AE to UE. The accuracy of the output file will increase with the number of points, as will the size of the data file. It should be noted that for SNOMAN, the difference between setting NALE to 100 and 700 is undetectable in the range of the electron and the amount of Čerenkov radiation produced.

1.4 DECK

This is the instruction to PEGS to write the information to file.

2 Running the Code

The instructions contained herein suppose that the user is working on a VMS machine, with a directory [*username*.snoman.pegs] and the following subdirectories:

.DATA contains the PEGS4 input data files, plus a copy of the PEGS4 fortran and MORTRAN.

.INP Contains the input files, as discussed above.

.LIST Contains the output listing files. These are generally no real use unless something has gone wrong.

.OUTPUT Contains the output data files.

The tools discussed below are designed to run assuming a directory structure as described, and will run on any account, provided the SNOMAN mnemonics are correctly set up. As an example the PEGS4 code is set up on AV3 in the directory [SUDBURY.SNOMAN.PEGS].

2.1 Generating a single file

Users running under VMS can use the file PEGS4.COM. This command file accepts one argument, which should be the name of the file containing the input data. ie if the input file is *myfile*.PEGS4INP, type @PEGS4 *myfile*. This will produce a file called *myfile*.DAT with the cross sectional data, and a file called *myfile*.LIS containing the running commentary from PEGS. UNIX users are on their own at this point, though if anybody wants to write a script file and add it to the collection, they are welcome to do so!

PEGS4.COM is set up to run in the directory [*username*.snoman.pegs], and takes its input data files from a [.data] subdirectory. Your input file should be placed in the [.INP] directory.

2.2 Generating the SNOMAN PEGS file

Add whatever input files are required to the [.INP] directory, remembering to give them a name along the lines of SNO*aa_description*.PEGS4INP. Here, *aa* is the next number in the series, and *description* is a mnemonic description of the material (use the current set of files as an example). Then run RUN_PEGS4.COM; this will build the complete SNOMAN data file and leave it in the [.DATA] directory. You might want to do this over coffee.

3 Comments

3.1 FORTRAN and MORTRAN

PEGS4 is written in MORTRAN, a preprocessor for FORTRAN. However, as MORTRAN is difficult to use, and may not be available on all systems, the FORTRAN version of the code is assumed throughout this document. The FORTRAN code should compile on all platforms with no warnings. The one exception is that some compilers will complain about HOLLERITHs split across a line break. The fix is to edit the code to avoid this.

3.2 Bugs and Fixes

Several minor problems have been noted. Some are oversights, some rely on aspects of the computer involved.

- Removal of 'write' to unit 10. Mandated by the fact that otherwise the file on unit 10 ran to megabytes. The write statement was a report from an integration routine.
- There is a compiler dependant bug in SPINIT. The problem is that the second material can cause an obscure error. The code complains about an incorrect user override of certain parameters, even if the user has done no such thing. This is a machine dependant feature which depends on whether the compiler has allocated memory to these variables during compile time. If they are static variables, then the values from the last call to SPINIT are still present and the code gets confused. Given that the code then overwrites these values, this is not a serious flaw!

  The two solutions are to remove the STOP statement in SPINIT, or to process multi file jobs one at a time, which is what the tools discussed above do.
- Error in NAMELIST/INP/. NALG was incorrectly specified.
- Change to array sizes. The arrays AFE, BFE, AFG, BFG were increased in size to 800 in order to accomodate larger values of NALE and NALG.

  Warning: PEGS does not check that it's array sizes are consistent with the requested size, and will cheerfully overwrite all sorts of memory if the requested size is bigger than the available array size. Either check that the arrays are large enough or use a compiler with a bounds checker.

References

[1] Walter R. Nelson, Hideo Hirayama and David W.O. Rogers, (1985) *The EGS4 code system*, SLAC report 265.