PSG2 / Serpent - a Quick Installation Guide

Below is a short description on the installation of the PSG2 / Serpent Monte Carlo reactor physics burnup calculation code. The procedure consists of three steps:

1. Installing the latest update
2. Compiling the source code
3. Installing the data libraries

The code has been developed and tested in PC Linux and MAC OS X operating systems and installation in other platforms may require additional steps.

Compiling Serpent

Serpent is written in the C programming language and compiled using the standard Make utility. The compilation should look something like:

```
~/src/serpent$ make
gcc -Wall -ansi -pedantic -ffast-math -O3 -c aburnoutput.c
gcc -Wall -ansi -pedantic -ffast-math -O3 -c addbuf.c
gcc -Wall -ansi -pedantic -ffast-math -O3 -c addburnisotopes.c
...
gcc -Wall -ansi -pedantic -ffast-math -O3 -c zaitoiso.c
gcc aburnoutput.o addbuf.o addburnisotopes.o ... zaitoiso.o  -lm  -lgd
-o sss
~/src/serpent$
```

The compilation should not result in any error or warning messages and it should produce an executable named “sss”.

GD Graphics library

The installation in some platforms may require changes in some of the compiler options in the Makefile. In particular, if the GD open source graphics library[1] is not available in the system, the

```
LDFLAGS += -lgd
```

[1] See: http://www.libgd.org/

line must be removed from the Makefile and line

```
CFLAGS += -DNO_GFX_MODE
```

added.

Debugging mode

The code can be compiled in the debugging mode by adding line

```
CFLAGS += -DDEBUG
```

in the Makefile. This activates various pointer and value checks during the calculation. The code runs slower, but errors are more likely to be spotted before inducing unexpected results.

It is recommended that the code is compiled in the debugging mode when new or poorly tested features are used. In case of crash or any unexpected behavior, the best way to proceed is to recompile the code in the debugging mode and repeat the calculation using the same random number seed (the “-replay” command line option).

Parallel calculation using MPI

Parallel calculation using MPI[2] also requires changes in the Makefile. The calculation mode is activated by adding line

```
CFLAGS += -DMPI
```

The compiler needs to know where to look for the associated libraries. In some installations this can be accomplished simply by changing the compiler from “`gcc`” to “`mpicc`”.

Making backups

A time-stamped backup can be made using the “bk” -option:

```
~/src/serpent$ make bk
zip "‘date +’backup/serpent\_%y%m%d%H%M.zip’‘" *.c *.h Makefile sss
  adding: aburnoutput.c (deflated 76%)
  adding: addbuf.c (deflated 55%)
  adding: addburnisotopes.c (deflated 80%)
```

[2] See: http://www-unix.mcs.anl.gov/mpi/

```
...
  adding: sss (deflated 53%)
cp sss "‘date +’backup/serpent\_%y%m%d%H%M’‘"
cp "‘date +’backup/serpent\_%y%m%d%H%M.zip’‘" ./serpent.zip
~/src/serpent$
```

The source code is zip-compressed in the “backup” subdirectory, which must be found in the source directory. The executable is copied in the same directory and renamed using the same time stamp, so that earlier versions can be called without re-compiling the source code.

Installing updates

Code updates are sent to registered users in zip-compressed format by e-mail.[3] The installation of updates is carried out by overwriting the existing source files:

```
~/src/serpent$ make bk
...
~/src/serpent$ unzip -o sssupX.Y.Z.zip
...
~/src/serpent$ make clean
...
~/src/serpent$ make
...
~/src/serpent$
```

It is always good practice to make a backup of the source code before installing the update. It is also important to realize that any modifications in the source code may be lost when the updated files are installed.

The updates are cumulative in the sense that update “sssupX.Y.Z+1.zip” contains all the modifications in update “sssupX.Y.Z.zip” and earlier.

Code version

The code version can be checked after the compilation using the “-version” command line option:

```
~/src/serpent$ ./sss -version
```

[3] Registration is accomplished by sending e-mail to: Jaakko.Leppanen@vtt.fi

```
  _                   .-=-.          .-==-.
 { }      __        .’ O o ’.       /  -<’ )--<
 { }    .’ O’.     / o .-. O \     /  .---‘
 { }   / .-. o\   /O  /   \  o\   /O /
  \ ‘-‘ /   \ O‘-’o  /     \  O‘-‘o /
   ‘-.-‘     ’.____.’       ‘.____.’

PSG2 / Serpent

A Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code

 - Version 1.1.7 (November 6, 2009) -- Contact: Jaakko.Leppanen@vtt.fi

 - Parallel calculation mode available

 - Geometry and mesh plotting available

~/src/serpent$
```

This also prints information about the availability of parallel calculation and geometry and mesh plotting.

Setting up cross section libraries

The code reads continuous-energy cross sections from ACE format data libraries. The directory file is different from the “xsdir” file used by MCNP and the conversion between the two formats is made using the “xsdirconvert” perl script:

```
~/xsdata/jeff31$ xsdirconvert.pl sss_jeff31.xsdir > sss_jeff31.xsdata
~/xsdata/jeff31$
```

The output is forwarded to a new file, which is read by setting the “acelib” parameter in the input (see the Manual).

The conversion script checks that each library file in the original xsdir exists and it is important that the first line in the file defines the correct data path. If the data is found in another location, the xsdir file must be edited before running the script. If all file paths are set correctly, the xsdir file should be directly usable by MCNP and the conversion to Serpent format should work without problems.

Two entries are written for each nuclide, one using the standard MCNP convention (ZA.id) and another one using the element symbol and the isotope mass (e.g. 92235.03c and U-235.03c for U-235). Either name can be used to identify the nuclide in the material compositions.

Isomeric states

The script assumes that nuclides in isomeric states are denoted by setting the third digit in the ZA to 3 (e.g. 61348 for Pm-148m or 95342 for Am-242m). If other convention is used, the isomeric state number (5. entry) must be set manually.

Data libraries provided with source code

Installation package 1.1.0 (NEA base version) contains ACE format cross section libraries based on the JEF-2.2, JEFF-3.1, ENDFB/VI.8 and ENDF/B-VII evaluated nuclear data files. The libraries are complemented with JEFF-3.1.1 data in package 1.1.7 (RSICC base version). The new libraries also include unresolved resonance probability table data, which is missing from the earlier version.

Standard ENDF format radioactive decay and fission yield libraries are provided for the same evaluations. All data is zip compressed and must decompressed before the installation.

File I/O

Data libraries read by Serpent may contain several hundred megabytes of data. It is therefore recommended that the libraries are installed in the system in such way that the data can be accessed directly. File servers may form bottlenecks that slow down the initialization phase of the calculation, which is reflected in the overall running time as well.