3.9.6

Abstract programming interface for MCTAL files

Nicolò Borghi, Konstantin Batkov

European Spallation Source, Tunavägen 24, 223 63, Lund, Sweden

E-mail: konstantin.batkov@esss.se

Abstract. MCNPX allows to save its calculation results in the MCTAL files and plot them by means of the MCPLOT utility. We have developed an implementation of abstract programming interface (API) to read MCTAL files. This API allows to extract data from the MCTAL files for further analysis and, if necessary, save it in any desired format. As an example, a converter from MCTAL to ROOT format is presented.

1. Introduction

The current work aims to provide a convenient mechanism to extract data from the MCNPX [1] output files in the MCTAL format. This mechanism is implemented as an abstract programming interface (API) developed in the Python language. It provides a set of classes to read MCTAL files and extract data from them in a fully automatic way. This possibility is essential for extensive MCNPX calculations, where one needs to analyse results of numerous runs.

The API manual and examples can be found on the project home page:
https://code.google.com/p/mc-tools

Interfacing with the mctal API does not require complex coding. It is as simple as importing the module, making an instance of the MCTAL class and calling the Read() method to get a list of all the tallies in the mctal file:

```
from mctal import MCTAL
m = MCTAL(mctalFileName)
tallies = m.Read() # get list of tallies from a mctal file
for f in tallies:
    ... # do your analysis here
```

2. ROOT converter

From our point of view, a big inconvenience of the standard MCPLOT utility is lack of ability to analyse data.

The ROOT framework [2, 3] developed at CERN is a tool to perform statistical data analysis. We use it in our everyday studies, therefore as an example of application of this API, we have developed a converter from the MCTAL format into the ROOT format.

```
[kbat@jasper ]$ mctal2root -h
usage: mctal2root [-h] [-v] mctal_file [root_file]

A mctal to ROOT conversion script.

positional arguments:
 mctal_file     the name of the mctal file to be converted
 root_file      the name of the output ROOT file

optional arguments:
 -h, --help     show this help message and exit
 -v, --verbose  explain what is being done
```

3. An example

Let's convert a MCTAL file into the ROOT format:

```
[kbat@jasper ]$ mctal2root mctal
  ROOT file saved to: mctal.root
[kbat@jasper ]$ ls -l mctal*
-rw-r----- 1 kbat 126510  mctal
-rw-r----- 1 kbat  21893  mctal.root
```

Note the size difference due to ASCII to binary conversion and the ROOT internal compression.

The ROOT file contains 4 objects which correspond to the MCNPX tallies:

```
[kbat@jasper ]$ root mctal.root
root [1] .ls
TFile**         mctal.root
 TFile*         mctal.root
  KEY: THnSparseT<TArrayF>      f6;1
  KEY: THnSparseT<TArrayF>      f15;1
  KEY: THnSparseT<TArrayF>      f25;1
  KEY: THnSparseT<TArrayF>      f35;1
```

Repeating the MCTAL file structure, the ROOT data is saved as a 8-dimensional histogram THnSparse:

```
root [1] f5->Print("a")
THnSparseT<TArrayF> (*0x2eb0250): "f5" ""
  8 dimensions, 1.0054e+06 entries in 1005402 filled bins
    axis 0 "f": 1 bins (0..1), variable bin sizes
    axis 1 "d": 1 bins (0..1), variable bin sizes
    axis 2 "u": 1 bins (0..50006), variable bin sizes
    axis 3 "s": 1 bins (0..1), variable bin sizes
    axis 4 "m": 1 bins (0..1), variable bin sizes
    axis 5 "c": 1 bins (0..1), variable bin sizes
    axis 6 "e": 201 bins (0..3000), variable bin sizes
    axis 7 "t": 5002 bins (0..1e+37), variable bin sizes
```

This histogram can be analysed and visualised by making use of standard ROOT techniques. For instance, let's plot the time spectrum:

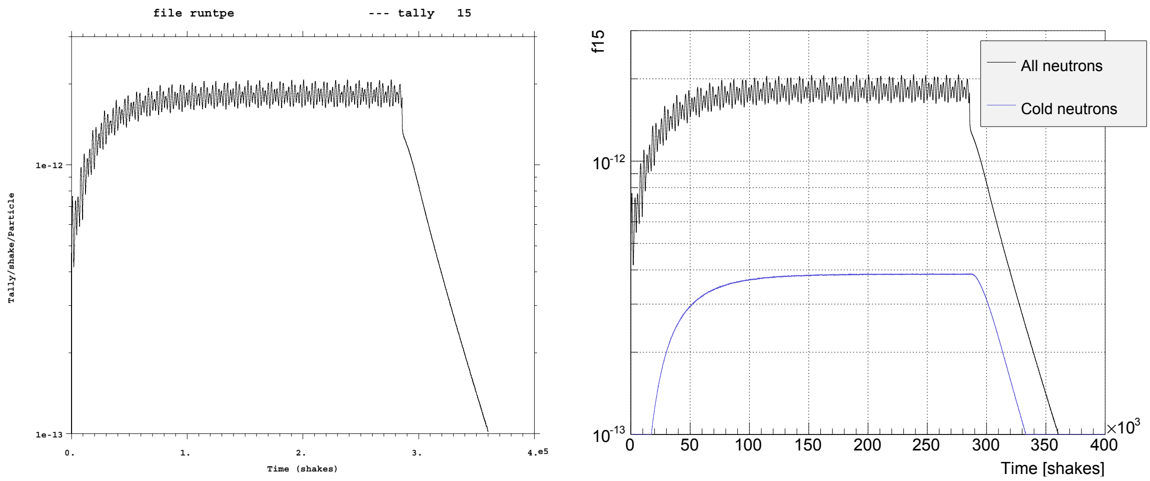

Figure 1. Examples of a histogram depicted with `mcplot` (left) and `ROOT` (right)

```
root [2] TH1 *h = f5->Projection(7) // axis 7 corresponds to
                                    // time
root [2] h->Scale(1, "width")       // divide the bins by their
                                    // width
root [3] h->Draw()                  // draw the histogram
```

The resulting distribution is shown in black in the right pad of Fig. 1.

Let's plot the time spectrum for the cold neutrons only (energy below 5 meV):

```
f15->GetAxis(6)->SetRangeUser(0, 5E-9) // select the energy range
TH1 *hc = f15->Projection(7)
hc->SetLineColor(kBlue)                // set the histogram color
hc->Scale(1, "width")
hc->Draw("hist,same")                  // draw at the same pad
```

The resulting spectrum is shown in blue in the right pad of Fig. 1.

`ROOT` allows to perform any kind of data analysis with these objects. Refer to the http://root.cern.ch website for details.

4. Supported and unsupported features

The code has been extensively used during last year by several persons, and we currently believe that is can handle all `mctal` records except the following ones:

- `KCODE` data;
- Tallies with perturbation records (produced by the `pert` card).

These features are not implemented just because the authors never use them, but if necessary they can be supported by the code in the future.

5. Conclusions

- We have developed an API to provide a convenient access to the information stored in the MCTAL files.
- Based on this API we have developed a tool to convert the MCTAL files into the ROOT file format.
- These tools simplify the visualisation and analysis of the results obtained with MCNPX.

6. See also

The MCTAL API and the ROOT converter are the parts of a larger project called mc-tools: https://code.google.com/p/mc-tools. This project includes several other Monte Carlo tools listed below:

- Converter of wssa files into the ROOT format;
- Converter of ANGEL files (standard PHITS output) into the ROOT format;
- A tool to animate 3D geometry in PHITS.

Acknowledgments

We thank Troels Schönfeldt, Esben Klinkby, Riccardo Bevilaqua and Furutaka Kazuyoshi for their discussions and contribution in the project.

References

[1] LANL Los Alamos, New Mexico, USA *MCNP5/MCNPX manual*

[2] Brun R and Rademakers F 1996 ROOT — an object oriented data analysis framework *Proceedings AIHENP'96 Workshop* (Lausanne, Switzerland) see also http://root.cern.ch.

[3] Brun R and Rademakers F 1997 *Nucl. Instrum. Methods Phys. Res., Sect. A* **389** 81–86