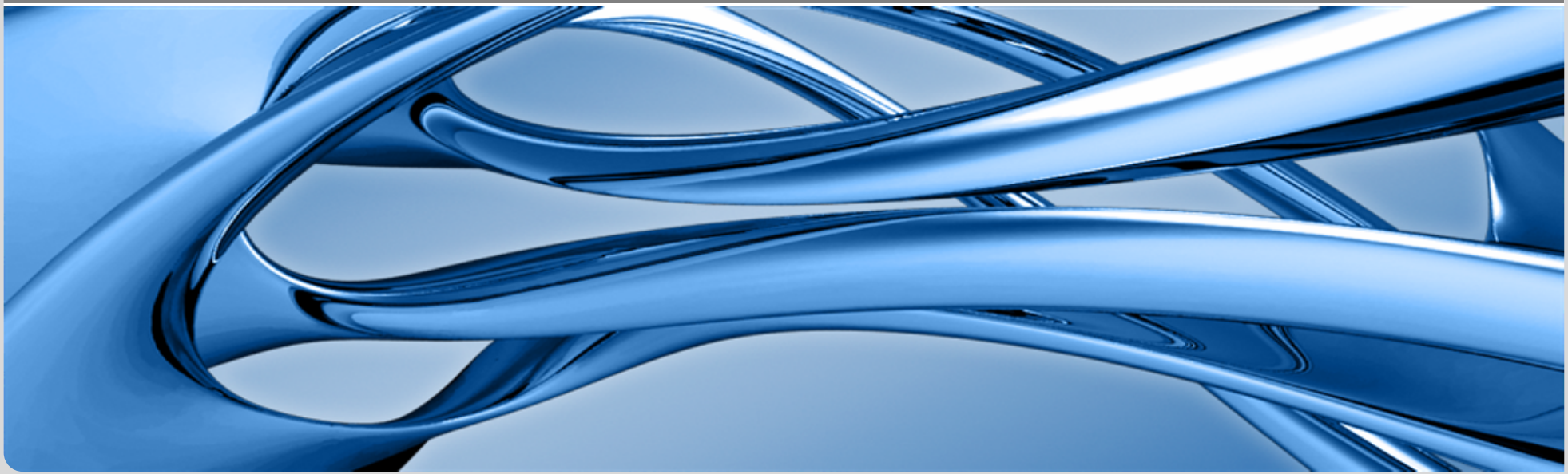
Karlsruhe Institute of Technology

Python Command Line Tool to Rename Cells, Surfaces, Universes and Materials in MCNP Input Files

A. Travleev,
Institute for Neutron Physics and Reactor Technology (INR), KIT

X^{th} ITER Neutronics Meeting, Cadarache, 29.06 – 03.07.2015



Karlsruhe Institute of Technology

Introduction

Problem statement

Approach to create complex MCNP model: develop parts, check, integrate.

This approach is followed, for example, to develop the ‘C-Lite’ MCNP model of ITER, where semi-detailed representation of the component latest designs is developed by several organizations.

Approach is straightforward, but:

- error-prone: some operations can be done only manually,
- tedious: model preparation can require several iterations

What modifications are needed?

MCNP input files undergo several modifications before can be integrated into common model:

- Add universe number to “real world” cells, e.g. `u=5`
- find free number ranges in the common model
- rename cells, surfaces, materials etc.
- wrap long lines, if necessary

Our tool:

To facilitate this otherwise tedious work, we have developed a tool with following features:

- Preserve input file structure, including comments and line wrapping. Introduced changes can be controlled by generic tools, like `diff` or `vimdiff`.
- Different mapping rules can be applied to different ranges of numbers. Thus, one can change only a subset of numbers, or ultimately specify new numbers to each cell/surface/material.
- If necessary, lines in the renumbered input file can be wrapped to obey the 80 characters limit.
- A log file containing all substitutions can be used as an input to perform reverse change. This is useful to verify results: the input file obtained by the reverse renumbering should be identical to the original input file.
- The tool is written in Python and is cross-platform. Its command line interface can be run on a remote machine with terminal access and does not require Python knowledge.

Karlsruhe Institute of Technology

Examples

Specify renaming rules in command line

In the command line one can specify values to be added to cell, surface, universe and material numbers.

```
numjuggler -c 10 -s 5 -m 100 -u -2   orig.inp > r1.inp
```

- `numjuggler` is the tool name,
- `orig.inp` is the original input file
- `r1.inp` is the modified input file

Result

orig.inp:

```
MCNP input file
c Envelope
10 0 -10 fill=5    $ model
20 0  10           $ other world
c interior
1   1 -17.0 -2     u=5
2   10 -1.0  2 -1 u=5
3   0         2  1 u=5

c surfaces
10 so 9
1 pz 0
2 py 1

c data cards
m1 $ enriched uranium
        92235.60c 90.0
        92238.60c 10.0
m10 $ water
      1001 2.0 8016 1.0
c  borated!
      5010 0.005
mt10 lwtr.01t
f4:n 10 (1  2 )
```

r1.inp:

```
MCNP input file
c Envelope
20 0 -15 fill=3    $ model
30 0  15           $ other world
c interior
11   101 -17.0 -7     u=3
12   110 -1.0  7 -6 u=3
13   0         7  6 u=3

c surfaces
15 so 9
6 pz 0
7 py 1

c data cards
m101 $ enriched uranium
        92235.60c 90.0
        92238.60c 10.0
m110 $ water
      1001 2.0 8016 1.0
c  borated!
      5010 0.005
mt110 lwtr.01t
f4:n 20 (11  12 )
```

- Input file structure is preserved, when possible
- Cells and surfaces are recognized in some data cards (see f4 tally)

Number by appearence in input file

Special syntax can be used to specify that elements (cells, surfaces, materials) are numbered by increasing numbers, in the order they appear in input file.

```
numjuggler -c i -s i -m i -u i       orig.inp > r2.inp
```

Result

orig.inp:

```
MCNP input file
c Envelope
10 0 -10 fill=5   $ model
20 0  10          $ other world
c interior
1   1 -17.0 -2    u=5
2   10 -1.0  2 -1 u=5
3   0        2  1 u=5

c surfaces
10 so 9
1 pz 0
2 py 1

c data cards
m1 $ enriched uranium
        92235.60c 90.0
        92238.60c 10.0
m10 $ water
      1001 2.0 8016 1.0
c  borated!
      5010 0.005
mt10 lwtr.01t
f4:n 10 (1  2 )
```

r2.inp

```
MCNP input file
c Envelope
1  0 -1  fill=1   $ model
2  0  1          $ other world
c interior
3   1 -17.0 -2    u=1
4   2  -1.0  2 -3 u=1
5   0        2  3 u=1

c surfaces
1  so 9
3 pz 0
2 py 1

c data cards
m1 $ enriched uranium
        92235.60c 90.0
        92238.60c 10.0
m2  $ water
      1001 2.0 8016 1.0
c  borated!
      5010 0.005
mt2  lwtr.01t
f4:n 1  (3  4 )
```

Get information

Command line option `--mode info`: inquire information about what number (ranges) are already used (Default mode is `--mode renum`, therefore not specified in previous examples).

```
numjuggler --mode info                    orig.inp > r3.txt
```

Result

`r3.txt`:

```
---------------------------------------- cel 5
         c  1 -- 3                              3
         c  10                                  1        6
         c  20                                  1        9
---------------------------------------- sur 3
         s  1 -- 2                              2
         s  10                                  1        7
---------------------------------------- mat 3
         m  0 -- 1                              2
         m  10                                  1        8
---------------------------------------- u 1
         u  5                                   1
---------------------------------------- tal 1
         t  4                                   1
---------------------------------------- tr 0
```

Its output not an input file, but text containing:

- Total amount of cells, surfaces etc.
- list of ranges and single numbers used in input
- Free slot length between ranges

Specify renaming rules in external file

Karlsruhe Institute of Technology

Renaming rules can be read from an external file with `--map` option.

Opposite to the `-c`, `-s`, `-m` and `-u` options, in the external file one can specify **different** rules for **different** ranges. Ultimately, all new cell, surface, material or universe names can be given explicitly.

```
numjuggler --map map4.txt              orig.inp > r4.inp
```

`map4.txt`:

```
# cells
          c  1 -- 3: +5     # add 5 to cells 1 to 3
          c 10 --20: +10    # add 10 to cells 10 to 20
# surfaces
          s  1 -- 2: 5      # 1 -> 5, 2 -> 6 etc
          s         :10     # add 10 to all other
# universes
          u 0: 1
          u 5: +1
# tallies not implemented yet
          t: 10
```

- output of `--mode info` option (previous slide) can be used as template for this file.
- Format detailed description by `numjuggler -h map`

Results

orig.inp:

```
MCNP input file
c Envelope
10 0 -10 fill=5    $ model
20 0  10           $ other world
c interior
1   1 -17.0 -2     u=5
2   10 -1.0  2 -1 u=5
3   0         2  1 u=5

c surfaces
10 so 9
1 pz 0
2 py 1

c data cards
m1 $ enriched uranium
        92235.60c 90.0
        92238.60c 10.0
m10 $ water
      1001 2.0 8016 1.0
c  borated!
      5010 0.005
mt10 lwtr.01t
f4:n 10 (1  2 )
```

r4.inp

```
MCNP input file
c Envelope
20 0 -20 fill=6    $ model
30 0  20           $ other world
c interior
6   1 -17.0 -6     u=6
7   10 -1.0  6 -5 u=6
8   0         6  5 u=6

c surfaces
20 so 9
5 pz 0
6 py 1

c data cards
m1 $ enriched uranium
        92235.60c 90.0
        92238.60c 10.0
m10 $ water
      1001 2.0 8016 1.0
c  borated!
      5010 0.005
mt10 lwtr.01t
f4:n 20 (6  7 )
```

- Some elements are not implemented yet (tallies, transformations)
- Implicit zero universe is not changed (see next)

Add explict `u=0`

Option `--mode uexp` adds explicit `u=0` to cells without `u` keyword.

This can be useful when combining several input files into one model using universes. When cells have explicit `u=0` entries, they can be renumbered in usual (see previous slides) ways.

```
numjuggler --mode uexp                    orig.inp > r5.inp
```

Result

orig.inp:

```
MCNP input file
c Envelope
10 0 -10 fill=5   $ model
20 0  10          $ other world
c interior
1   1 -17.0 -2    u=5
2   10 -1.0  2 -1 u=5
3   0        2  1 u=5

c surfaces
10 so 9
1 pz 0
2 py 1

c data cards
m1 $ enriched uranium
        92235.60c 90.0
        92238.60c 10.0
m10 $ water
      1001 2.0 8016 1.0
c  borated!
      5010 0.005
mt10 lwtr.01t
f4:n 10 (1  2 )
```

r5.inp

```
MCNP input file
c Envelope
10 0 -10 fill=5   u=0 $ model
20 0  10          u=0 $ other world
c interior
1   1 -17.0 -2    u=5
2   10 -1.0  2 -1 u=5
3   0        2  1 u=5

c surfaces
10 so 9
1 pz 0
2 py 1

c data cards
m1 $ enriched uranium
        92235.60c 90.0
        92238.60c 10.0
m10 $ water
      1001 2.0 8016 1.0
c  borated!
      5010 0.005
mt10 lwtr.01t
f4:n 10 (1  2 )
```

Wrap long lines

Option `--mode wrap` wraps lines in the MCNP input file to fit the 80-chars limit.

Only meaningful line parts are wrapped: if a line exceed 80 characters due to comments (any entries after “$” or “&”), it is not wrapped.

```
numjuggler -s 10000 -c 100000          or2.inp > r6L.inp
numjuggler --mode wrap                 r6L.inp > r6s.inp
```

`or2.inp`:

```
MCNP input file
10 1 -10.0 -1 -2 -3 4 5 6 -7 (-8:-9:10) imp:n=1 imp:p=1   $ model
20 0  10                                                  $ other world

c surfaces
10 so 9
1 pz 0
2 py 1

c data cards
f4:n 10 (1 2 3 4 5 6 7 8 9 10)
```

Result

After renaming, file `r6L.inp`:

```
MCNP input file
100010 1 -10.0 -10001 -10002 -10003 10004 10005 10006 -10007 (-10008:-10009:10010) imp:n=1 imp:p=1   $ model
100020 0  10010                                                                    $ other world

c surfaces
10010 so 9
10001 pz 0
10002 py 1

c data cards
f4:n 100010 (100001 100002 100003 100004 100005 100006 100007 100008 100009 100010)
```

After wrapping, file `r6s.inp`:

```
MCNP input file
100010 1 -10.0 -10001 -10002 -10003 10004 10005 10006 -10007 $ model
      (-10008:-10009:10010) imp:n=1 imp:p=1
100020 0  10010                                                $ other world

c surfaces
10010 so 9
10001 pz 0
10002 py 1

c data cards
f4:n 100010 (100001 100002 100003 100004 100005 100006 100007 100008
      100009 100010)
```

Note, wrapping is not done automatically, user must specify `--mode wrap` in a separate run (to preserve input file structure for comparison)

Reverse renumbering

The `--log` option generates log file where all number substitutions are stored. It can be used as a map file to perform reverse renaming.

This can be used to verify the tool: input after reverse renaming must be equal to the original one.

```
# generate log
numjuggler -c 10 --log log7.txt      orig.inp > r7.inp
# apply log as map file
numjuggler --map log7.txt            r7.inp > rev7.inp
```

`log7.inp`:

```
------------------------------------------------------------------------------
c     11:        1
c     12:        2
c     13:        3
c     20:       10
c     30:       20
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
```

Results

orig.inp:

```
MCNP input file
c Envelope
10 0 -10 fill=5    $ model
20 0  10           $ other world
c interior
1    1 -17.0 -2     u=5
2    10 -1.0  2 -1 u=5
3    0         2  1 u=5

c surfaces
10 so 9
1 pz 0
2 py 1

c data cards
m1 $ enriched uranium
        92235.60c 90.0
        92238.60c 10.0
m10 $ water
      1001 2.0 8016 1.0
c  borated!
      5010 0.005
mt10 lwtr.01t
f4:n 10 (1  2 )
```

r7.inp:

```
MCNP input file
c Envelope
20 0 -10 fill=5    $ model
30 0  10           $ other world
c interior
11    1 -17.0 -2     u=5
12    10 -1.0  2 -1 u=5
13    0         2  1 u=5

c surfaces
10 so 9
1 pz 0
2 py 1

c data cards
m1 $ enriched uranium
        92235.60c 90.0
        92238.60c 10.0
m10 $ water
      1001 2.0 8016 1.0
c  borated!
      5010 0.005
mt10 lwtr.01t
f4:n 20 (11  12 )
```

rev7.inp:

```
MCNP input file
c Envelope
10 0 -10 fill=5    $ model
20 0  10           $ other world
c interior
1    1 -17.0 -2     u=5
2    10 -1.0  2 -1 u=5
3    0         2  1 u=5

c surfaces
10 so 9
1 pz 0
2 py 1

c data cards
m1 $ enriched uranium
        92235.60c 90.0
        92238.60c 10.0
m10 $ water
      1001 2.0 8016 1.0
c  borated!
      5010 0.005
mt10 lwtr.01t
f4:n 10 (1   2  )
```

Karlsruhe Institute of Technology

Concluding remarks

Potential application to C-Lite

- Extreme scenario:
  - Distribute C-lite as empty envelopes together with a set of model parts of several complexity. User builds dedicated model, whith detailed model parts in regions of interest and with simplified model parts in other regions.
  - Get rid of numbering policies: let user decide how elements are numbered.
  - Requires further development
- Moderate scenario:
  - Apply for model integration, e.g. in current and upcoming task orders
  - mature enough, but lacks user feedback

Binding to Python

- Classes and functions defined in `numjuggler` package can be used in Python scripts: mapping rules can be arbitrarily complex.
- MCNP input file parser can be used for other purposes

Availability

On github, github.com/inr-kit/numjuggler, one can find:

- Source files for further development, or
- Distribution package to install and use, and
- Installation instructions.

Thank you for your attention!