Algorithm for optimization of MCNP geometry input.

Report

Patrick Sauvan

Abstact:

In this work a tool for the optimization of the MCNP geometry in term of memory requirement is developed. The method used to perform the optimization is based on the manipulation of the MCNP cell definition. It consists of the elimination of the complementary cell operators and the reduction of the size of the cell definition by removing the redundant parentheses.

The MCNP input file optimization process is carried out using an ad-hoc modified version of the Numjuggler code in which removing of redundant parentheses and cell complementary operators function have been implemented.

1. Introduction.

The size of the memory required by MCNP to store the geometry depends on two terms [1], the total number of "words" (a work is any element in the cell definition "(", ")", ":" or surface number) in the input file and the total number of complementary cell operators (# operators). The elimination of these operators and redundant parentheses can reduce significantly (from 20 to 60 %) the memory required by MCNP to handle the geometry definition.

In this work, algorithms to remove complementary operators and redundant parentheses have been implemented in the numjuggler code. Numjuggler is a tool developed to rename cells, surfaces, materials and universes in MCNP input files. This tool is very useful when working with huge MCNP model like Cmodel. New functions have been added to numjuggler to carry out the optimization of the MCNP geometry.

Three capabilities have been implemented in the last version of numjuggler [2] (2.23a.17), these capabilities are:

- Removing of the complementary cell operators (# operators),
- Removing redundant parentheses,
- Information of the estimated memory requirement to store the geometry in MCNP.

The modifications introduced in numjuggler have been implemented in a separated module which is not interacting with original numjuggler modules. Thus, these new modifications have no impact on the original numjuggler results. New implementations have been tested, showing that the processing of the MCNP input file doesn't change the geometry in which MCNP perform the transport calculation.

The new numjuggler functions are called using the same syntax as numjuggler original functions.

2. Removing complementary cell operator.

The command to remove the complementary cell operator is :

--mode remh

“--mode" is the original numjuggler mode to call the different functions. Here the new keyword is “remh” associated to the function which removes the complementary cell operator.

The user can track the modifications of the input by telling the script to write in a file the modifications carried out. The option is enabled with the command --log filename.

For example,

--mode remh --log remhLog

will write in the remhLog file all the cells in which the complementary cell operator(s) has(ve) been removed.

In Figure 1, an example of the output log file is shown. If complementary cell operators operate on a cell definition (e.g #(-3 : 4 5)) the cell definition is written in the file. If the operator refers to a cell number (e.g #458) the label of the cell in written in the file.

```
-------------------------------------------------
Cell     43055 :

     Complementary cell definition :
 1:   ((43274 43527 -42725 -43273))

-------------------------------------------------
Cell     47050 :

     Complementary cell definition :
 1:   ((47271 47451 -46723 -47270))

-------------------------------------------------
Cell     50712 :

     Complementary cell number :
 1:      50926

-------------------------------------------------
Cell     50864 :

     Complementary cell number :
 1:      50857
 2:      50860
```

Figure 1: Example of a logfile produced by the function remh.

The function remh remove all the complementary cell operators found in the MCNP input file. The function remove both operators referring to a specific cell (#cell_number) and to the operators applied to cell definition (e.g. #(-3:4 5)). The function removes also nested

complementary cell operators; i.e. a complementary cell of a cell containing a complementary cell operators.

However, the function is not able to remove a complementary operator referring to a transformed cell. In this case the complementary cell operator remains in the cell definition.

The procedure used to obtain the complementary definition of the cell, is the same as defined in the MCNP manual [3].

The steps are:

- The cell complementary cell is bracketed "(complementary cell)" ,
- Sign of surfaces is changed,
- Blank " " (intersections) are replaced by ":" and ":" are replaced by ")(",
- Redundant parentheses are removed.

In Figure 2 an example of the construction of the complementary cell is shown.

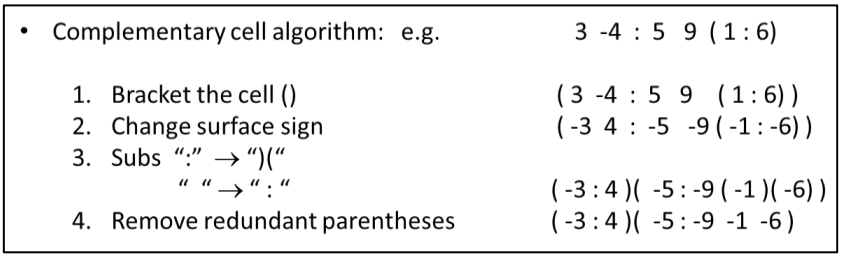


Figure 2: Step of the construction of the complementary cell operator.

3. Removing redundant parentheses.

The command to remove the redundant parentheses is :

--mode remrp

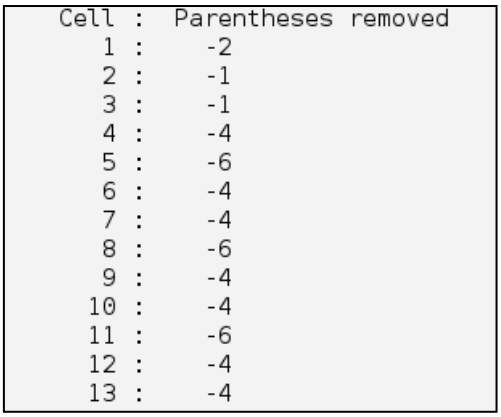
The new keyword "remrp" is associated to the function which removes the redundant parentheses.

The user can track the modifications of the input by telling the script to write in a file the modifications carried out. The option is enabled with the command --log filename.

For example,

--mode remrp --log remrpLog

will write in the remrpLog file the number of redundant parentheses removed in each cell. In Figure 2, an example of the output log file is shown. The couple "()" is counted as one redundant parentheses.



Cell	:	Parentheses removed
1	:	-2
2	:	-1
3	:	-1
4	:	-4
5	:	-6
6	:	-4
7	:	-4
8	:	-6
9	:	-4
10	:	-4
11	:	-6
12	:	-4
13	:	-4

Figure 3: Example of a logfile produced by the function remrp.

In MCNP the presence of parentheses or colon in the cell definition makes the cell considered as complicated. During transport simulation complicated cells are treated differently with respect simple cells in MCNP. The possibility to define a cell as simple or complicated only applies to cells defined with only intersection operators.

If the geometry is correctly defined, the result of the simulation should be independent of the nature of the cell (simple or complicated). But if there are errors in the geometry definition then the result may be different whether the cell is considered as simple or complicated.

During the process of removing the redundant parentheses, the cells with only surface intersections and defined as complicated (i.e. with parentheses in the definition) are changed to simple cells because in this case all parentheses are redundant.

To preserve or change the nature of the cell, an option has been added to the "remrp" function. This option allows selecting how the user wants to change or preserve the cell nature. Three options are available:

- **nochg** : the nature of the cell is preserved. One parenthesis is left on cells defined with only intersections if these cells were originally defined as complicated.
- **cc** : all cells are considered as complicated. One parenthesis is left on complicated cell definition and one parenthesis is added to cells originally defined as simple.
- **all :** remove all redundant parentheses independently of preserving the nature of the cell.

By default the "remrp" option is "nochg" . To enable another option the "-opt [option]" can be used in the command line.

numjuggler --mode remrp -opt all input > output

The "remrp" function considers as redundant parentheses the following case (bold red parenthesis):

- (**(** anything**)** :
- : **(** anything**)**)
- : **(** anything**)** :
- **(** A B only intersections … **)** where A, B, ... = (anything) or number

4. Information related to MCNP memory consumption.

The last function implemented in numjuggler provides the information on the estimated length of the geometry definition. This information includes the number of words and complementary cell operators used in the geometry definition, the length of the longest cell and the estimated size of the memory required by MCNP to store this geometry. A list of the cells with a complementary cell operator is also displayed after the memory information.

The information request is called in numjuggler with the minfo keyword:

numjuggler --mode minfo input

In figure 4, an example of the information provided is shown.

```
Total words            :  8304762
Total hash             :      340
Hashcel                :      260
Hashsurf               :       80
Longest cell           :      131
Words in longest cell  :     1090

MCNP estimation :
    mlja                          :   70778854
    Estimated memory requirement :       1.1GB
    %cell length, %number #      :       82.2%   17.8%

  Cell name    total #   cell #   surf #
      10290          1        0        1
      10291          1        0        1
      28833          1        1        0
      40706          1        1        0
      43055          1        0        1
      47050          1        0        1
      50712          1        1        0
      50864          2        2        0
      50927          1        1        0
      51012          1        1        0
```

Figure4: Information displayed by minfo function.

5. Verification process.

The correct implementation of the “remh” and “remrp” functions have been verified by two methods:

- In MCNP, lines have been added in the source code after the geometry processing, to write the elements of the lja array (array storing the MCNP geometry) in a file. These files produced by MCNP when the original input file and the optimized file are used are compared.

- Statistical volume calculations have been performed on the geometry using the original and optimized input files.

If the functions are correctly implemented, the comparisons in both tests should give exactly the same result. Both verification tests have been carried out successfully on Clite-R131031, Cmodel-R2.1-16121, and Cmodel-R171031 models. For all these models simulations give identical results.

6. Optimization performances.

The amount of memory saved after the file optimization is depending on the original model and mainly on the number of complementary cell operators defined in the input file. The elimination of the redundant parentheses, besides the memory optimization, reduce the time spend by MCNP in processing the input geometry. In the following table a summary of the performance of the optimization on the different models is shown.

		Original input	Remove #	Remove Redun. Parent.
C-lite R131031	# lja elements	2873960	1363446	1146524
	MCNP Memory	44MB	21MB	18MB
	%memory words	47%	100%	100%
	% memory #	53%	0%	0%
	Memory saving	0	53%	60%
	Initialization time	1.1 min	1.05 min	0.95 min
	Time Gain	0%	5%	14%
C-Model 2016_v1_R2.1	# lja elements	69354982	43918320	38420171
	MCNP Memory	1.0GB	670MB	586MB
	%memory words	63%	100%	100%
	% memory #	37%	0%	0%
	Memory gain	0	37%	45%
	Initialization time	203 min	225 min	94 min
	Time Gain	0%	-11%	54%
C-Model R171031	# lja elements	70778854	58260158	50501422
	MCNP Memory (GB)	1.1GB	889MB	771MB
	%memory words	82%	100%	100%
	% memory #	18%	0%	0%
	Memory gain	0	18%	27%
	Initialization time	320 min	345 min	128 min
	Time Gain	0 %	-8%	60%

Table1 : Input optimization performance.

7. References.

[1] J.Alguacil, P.Sauvan, R.Juarez, J.P.Catalan, "Assessment and optimization of MCNP memory management for detailed geometry of nuclear fusion facilities", Fusion Engineering and Design, DOI: 10.1016/j.fusengdes.2018.02.048, (2018)

[2] *NUMJUGGLER*, https://github.com/inr-kit/numjuggler

[3] X-5 Monte Carlo Team, "MCNP — A General Monte Carlo, N-Particle Transport Code, Version 5, Volume I: Overview and Theory ",LA-UR-03-1987, (2003)