Monte Carlo Parameter Studies and Uncertainty Analyses with MCNP5

Forrest B. Brown[1], Jeremy E. Sweezy[1], & Robert Hayes[2]

[1] *Diagnostics Applications Group (X-5), Los Alamos National Laboratory,*
PO Box 1663, MS F663, Los Alamos, NM 87545, <fbrown@lanl.gov>

[2] *Radiological Technology, MS 452-09, WIPP Site,*
PO Box 2078, Carlsbad NM 88220, <robert.hayes@wipp.ws>

A software tool called *mcnp_pstudy* has been developed to automate the setup, execution, and collection of results from a series of MCNP5 Monte Carlo calculations. This tool provides a convenient means of performing parameter studies, total uncertainty analyses, parallel job execution on clusters, stochastic geometry modeling, and other types of calculations where a series of MCNP5 jobs must be performed with varying problem input specifications.

KEYWORDS: *Monte Carlo, MCNP, parameter studies, uncertainty quantification*

1. Introduction

Monte Carlo codes are being used for a wide variety of applications today due to their accurate physical modeling and the speed of today's computers. In most applications for design work, experiment analysis, and benchmark calculations, it is common to run many calculations, not just one, to examine the effects of design tolerances, experimental uncertainties, or variations in modeling features. We have developed a software tool for use with MCNP5 [1] to automate this process. The tool, *mcnp_pstudy*, is used to automate the operations of preparing a series of MCNP5 input files, running the calculations, and collecting the results. Using this tool, parameter studies, total uncertainty analyses, or repeated (possibly parallel) calculations with MCNP5 can be performed easily. Essentially no extra user setup time is required beyond that of preparing a single MCNP5 input file.

2. Description of *mcnp_pstudy*

mcnp_pstudy permits a user to create input for MCNP5 using symbolic parameters (or variables) which may be obtained from a list of specified values, from random sampling of standard probability densities, or from mathematical operations performed using other variables. In addition, constraints can be defined in terms of inequality relations involving the variables and mathematical operations. These variables are then used to create a series of MCNP5 input files. In doing so, the variables may be combined using an "outer product" approach, where all possible combinations of all variables are used, or an "inner product" approach, where only corresponding entries from each variable specification are used. After creating the MCNP5 input files, the series of calculations can be run on the user's computer or submitted to a batch queuing system, and results from tally files can be collected after completion of the runs. The simplest application of *mcnp_pstudy*, varying just the random number seed for a series of

repeated calculations, can be used for running a set of parallel jobs on a cluster or parallel computer. The sections below describe the separate functions of *mcnp_pstudy*.

2.1 Parameter Definition

Symbolic parameters are defined in the MCNP input file using MCNP comment lines which begin with the special character string "c   @@@ ". The parameters may be upper-, lower-, or mixed case character strings including numbers and underscores. The value of each parameter may be defined as a single value, a list of values, the result of sampling from standard statistical distributions, or as the result of arithmetic statements that may include previously defined parameters. The general forms permitted are:

```
C   @@@     P1   =   value
C   @@@     P2   =   value1 value2 ... valueN
C   @@@     P3   =   normal   N  ave  dev
C   @@@     P4   =   uniform  N  min  max
C   @@@     P5   =  (  arithmetic-statement   )
C   @@@     P6   =   repeat  N
```

The particular values specified for forms P1 and P2 may be either numbers or character strings (without embedded blanks). For form P3, the keyword **normal** is followed by an integer number N denoting how many random samples to take from a normal (Gaussian) probability density with a mean of *ave* and standard deviation of *dev*. For form P4, N random samples are taken from a uniform density over the interval [*min*, *max*]. For form P5, standard arithmetic operators may be used (with ** used for exponentiation), along with ordinary functions such as *sin*, *cos*, *log*, *log10*, *exp, int*. In addition, the *rand(X)* function may be used to provide a random (real) number in the range (0,*X*), or (0,1) if X is omitted. The arithmetic expression may involve constants or previously-defined parameters. Parentheses may be used as needed, but the outmost parentheses are required. Form P6 can be used to force a calculation to be repeated N times. It is useful for repeating calculations when a single scalar variable is varied, such as the random number seed for a problem. Some examples of parameter definitions are:

```
MCNP title line
C
C   @@@   RADIUS = 8.741
C   @@@   U_DENSITY_STD =   18.74
C   @@@   FACT  = normal 25  1.0   .05
C   @@@   U_DEN = (  FACT * U_DENSITY_STD  )
```

2.2 Parameter Expansion and Evaluation

After parsing the entire MCNP input file for parameter definitions, *mcnp_pstudy* determines how many "cases" (i.e., MCNP jobs) are required to be run and the values of the parameters to substitute for each case. The parameters may be combined in an "outer" fashion where all possible combinations of all parameters are used, or in an "inner" fashion where successive values of each parameter are grouped together. The options -outer or -inner may be specified on the command line or within the input file using the OPTIONS= directive described below. For example, with the parameter specification

```
C   @@@   A = 1   2
C   @@@   B = 3   4
C   @@@   C = 5
```

specifying the outer product approach results in these case definitions:

```
Case 1:          A=1,  B=3,  C=5
Case 2:          A=2,  B=3,  C=5
Case 3:          A=1,  B=4,  C=5
Case 4:          A=2,  B=4,  C=5
```

In this example, parameters A, B, and C have respectively 2, 2, and 1 entries defined, so that the total number of cases for the outer expansion is 2x2x1=4. In expanding the parameter definitions into sets of parameters for each case, parameters are varied in order of their definition in the MCNP input file. Parameters with only a single value are simply used as-is for each case.

For the inner product approach, the total number of cases is the same as the length of the longest parameter list in the input. For parameters with a shorter list or single value, the last (or only) entry is repeated as needed to pad out the lists. For the example above, specifying the inner product approach would result in these 2 cases:

```
Case 1:          A=1,  B=3,  C=6
Case 2:          A=2,  B=4,  C=6
```

When a parameter definition includes either random sampling or arithmetic (forms P3, P4, P5 above), the parameter is expanded symbolically and then evaluated separately for each case. In this evaluation, previously defined parameters may be used, with their values taken for each specific case. For example, these parameter definitions

```
C  @@@  A = 4.5
C  @@@  B = normal 2  1.0 .05
C  @@@  C = ( A * B )
```

would be expanded for the outer product approach as

```
Case 1:  A=4.5,   B=N1,  C=4.5*N1
Case 2:  A=4.5,   B=N2,  C=4.5*N2
```

where *N1* and *N2* are independent random samples from a N(1.0,.05) normal distribution.

2.3 Constraints

During the process of expanding and evaluating the case parameters, constraints on the parameters may be specified. The general form for specifying a constraint is

```
C  @@@  CONSTRAINT = ( logical-expression )
```

where *logical-expression* is an expression which may include previously defined parameters, comparison operators ($<$, $<=$, $>$, $>=$, $==$, $!=$), and logical operators ($\&\&$, $\|$). Parentheses may be used as needed, but the outermost parentheses are required. As many constraint statements as desired may be supplied.

After the parameters for a case are expanded and evaluated, all of the constraint expressions are evaluated. If any of the constraint expressions are false, then all the parameters for that case are rejected and then reevaluated. This process is repeated until the case parameters satisfy all constraints. The use of constraints permits modeling such conditions as a dimension which varies according to a normal distribution, but does not exceed some maximum value.

2.4 Creation of Directories for Each Case

After expansion of the parameter definitions into sets that define each case, subdirectories are created to hold the MCNP input and output files for each case. We have assumed that the user is running in a Unix environment or in the Cygwin environment for Windows PCs. The resulting directory structure has the form

```
$JOBDIR/case001
$JOBDIR/case002
...
```

where `$JOBDIR` is a user-specified path (with a default of the current working directory) and `case001`, `case002`, etc., are subdirectory names. During the setup for each case, the case input file is placed in each of these subdirectories. For other needed files such as *wwinp, xsdir, rssa*, soft links are created in each of the subdirectories which point to the original files. Later during execution, the case output files are stored in these subdirectories.

2.5 Parameter Substitution and Case Creation

After expanding and substituting the parameters into cases, the MCNP input file is again parsed to perform parameter substitution and create a separate input file for each case. For each case, the input file lines beginning with "`c    @@@`" are discarded. Each line in the file is then examined to locate all occurrences of the defined parameters. The symbolic parameters are then replaced by their values for the case being prepared, and the resulting lines are written to a file named `inp` in the subdirectory for the case. Following these actions, each case subdirectory `case001`, `case002`, etc., will contain an MCNP input file named `inp` with parameter values unique to that case.

2.6 Running or Submitting All Cases

After the subdirectories and input files have been created for each of the cases, *mcnp_pstudy* can either run the MCNP jobs for each case on the current host computer in sequential order, or submit the jobs to a batch queuing system for a cluster or parallel computer. We have assumed that the batch queuing system is LSF with the `bsub` command, but it should be simple to convert to other batch systems. Options for the batch queuing system and for the individual MCNP5 jobs can be supplied to *mcnp_pstudy*, either via the command line or embedded in the MCNP input file on lines beginning with "`c   @@@  OPTIONS = `".

2.7 Collecting and Combining Results

After all of the MCNP5 jobs for each case have completed, *mcnp_pstudy* can be used to collect tally results from each case and average the results together (if appropriate). A related utility, *merge_mctal*, can also be used to merge the tally results from individual *mctal* files into a single combined file.

When *mcnp_pstudy* is used to perform repeated calculations of a problem, results from the repeated calculations can be combined using a batch-statistics approach. That is, for each tally quantity X, the series of results from the repeated calculations, $\{X_k, k=1,M\}$, where M is the number of repeated calculations, can be computed as:

$$\overline{X} = \frac{1}{M} \sum_{k=1}^{M} X_k$$

$$\sigma_{\overline{X}} = \sqrt{\frac{1}{M-1} \left[\frac{1}{M} \sum_{k=1}^{M} X_k^2 - \overline{X}^2 \right]}$$

(1)

For normal Monte Carlo calculations, where the input parameters (i.e., problem specifications) are fixed, the statistical uncertainty in the results decreases as $1/\sqrt{N}$, where N is the number of histories run in the problem. For example, to reduce the standard deviation in problem results by a factor of 10, the total number of histories must be increased by a factor of 100. For calculations performed using *mcnp_pstudy*, this same scaling behavior applies only when *mcnp_pstudy* is used to repeat a calculation without varying any of the problem input parameters (with the notable exception of the starting random number seed, which should be changed for each repetition of a problem). The total number of histories in a series of repeated calculations is N*M, where N is the number of histories in each single calculation and M is the number of repetitions. Running 100 separate calculations (each with a unique random number seed) will reduce the standard deviation by a factor of 10. Running an infinite number of histories or repeated calculations will reduce the standard deviation in computed results to zero.

However, when *mcnp_pstudy* is used to vary the problem parameters by sampling from a statistical distribution, such as using a normal distribution to statistically sample material densities or cell dimensions, the statistical uncertainty computed using Eq. (1) will not decrease as $1/\sqrt{N}$ when more histories or jobs are run. The variance in a tallied result can be decomposed into 2 components – a variance due to the random sampling which occurs during each particle history in the Monte Carlo, and a variance caused by the variation in problem input parameters or initial conditions:

$$\sigma_{\overline{X}}^2 = \sigma_{\overline{X},MC}^2 + \sigma_{\overline{X},IC}^2$$

(2)

As the number of histories N increases, $\sigma_{\overline{X},MC}$ will decrease as $1/\sqrt{N}$ and approach zero as infinite histories are run. However, the variance due to random sampling of the initial conditions will remain constant, independent of the number of histories run in the problem, $\sigma_{\overline{X},IC} = constant$.

3. Applications

Some of the principal applications of *mcnp_pstudy* are:

1. Parameter studies

   For example, if the minimum, average, and maximum values are specified for several dimensions in a problem, then *mcnp_pstudy* can create and run a series of MCNP5 calculations with all combinations of the dimensions. Any items in the normal MCNP5 input file can be varied in this way, including cross-section library identifiers for nuclides in the problem, material properties, and geometric information. A series of cases with varying control rod positions or

boron concentrations could be used for determining critical rod height or critical boron concentrations.

2. Determining the effect of uncertainties in material densities or geometric dimensions

   In modeling experiments or benchmarks, some problem parameters frequently have uncertainties. In using *mcnp_pstudy*, these parameters can be declared as variables to be sampled from standard probability densities, such as a normal or uniform density. A series of runs can be constructed and executed, with the results collected, followed by calculation of the average and standard deviation of tallied quantities. The reported standard deviations would take into account the effects of the uncertainties in the input quantities, in addition to the normal statistical uncertainties from the individual Monte Carlo runs. Thus, this tool could be used for direct quantification of the total uncertainty in a series of calculations.

3. Parallel processing using a "parallel jobs" paradigm

   Rather than running a single long calculation using parallel processing to reduce the run time, *mcnp_pstudy* can be used to set up and submit many separate jobs to a cluster. The jobs can execute concurrently (if there are sufficient available nodes in the cluster), and then *mcnp_pstudy* can be used to collect and combine the results. This type of parallelism, i.e. parallel jobs, can be very effective in utilizing Linux clusters, since execution of the jobs can begin as soon as any resources become available; there is no reservation and waiting time for a large number of nodes to be free. If many nodes are free, all of the jobs can execute concurrently. If only a few nodes are free, some jobs can execute immediately and others will follow as resources become available. Job parallelism can provide effective load-balancing on a cluster and help reduce scheduling difficulties typical of single parallel jobs using large numbers of processors.

4. Simulation of stochastic geometry

   There has been much recent interest in analyzing pebble bed reactors and other advanced reactor concepts where small fuel particles are located randomly. Such stochastic geometries can be created using *mcnp_pstudy*, with a separate stochastic realization for each case in a series of MCNP jobs. Collecting the results from a series of these cases would yield results which include the effects of the stochastic variations in geometry, giving larger, more realistic uncertainties on computed results.

4. Results

A simplified model of the Godiva bare-sphere critical experiment is used to provide simple examples of the application of *mcnp_pstudy*. Note that the uncertainties on physical quantities used herein are contrived, used for illustrative purposes only, and do not reflect the known uncertainties in the actual experiment. Figure (1) shows the original MCNP5 input file for the problem (gdv) and a modified input file (gdv-A) which directs *mcnp_pstudy* to generate a random odd integer starting seed for the MCNP5 random number sequence for each case. Using *mcnp_pstudy* and MCNP5 on a Linux cluster, 50 cases were run for gdv-A (with a different random seed for each case) in the same amount of time as the single calculation for problem gdv. The speedup due to job-parallel processing was 50X for this trial. In

```
gdv                                         gdv-A
1  -18.74   -1    imp:n=1                    C @@@   RNSEED = ( 2*int(rand(1000000))+1 )
2    0      1    imp:n=0                     C @@@   xxx    = REPEAT 50
                                             1  -18.74   -1    imp:n=1
1      so 8.741                              2    0      1    imp:n=0

kcode 10000  1.0   15   115                  1      so 8.741
ksrc  0 0 0
m1    92235 -94.73   92238 -5.27             kcode 10000  1.0   15   115
prdmp 0 0 1 1 0                              ksrc  0 0 0
                                             m1    92235 -94.73   92238 -5.27
                                             prdmp 0 0 1 1 0
                                             rand  seed=RNSEED
```

Figure 1. (left) MCNP input for simple Godiva calculation, (right) MCNP input for
mcnp_study, choosing a new (odd) random seed for each case

```
gdv-D                                        gdv-E
c vary fuel density - normal, 5%sd,          c vary fuel radius - normal, 5%sd
c adjust the radius to keep constant mass     c vary fuel density- normal, 5%sd
c                                            c
c @@@ OPTIONS = -jobdir GDV_D                 c @@@ OPTIONS = -jobdir GDV_E
c                                            c @@@ OPTIONS = -inner
c @@@ FACT= normal 50  1.0 .05                c
c @@@ UDEN= ( 18.74*FACT )                    c @@@ DFACT = normal 50  1.0 .05
c @@@ URAD= ( 8.741*(18.74/UDEN)**.333333 )   c @@@ UDEN = ( DFACT * 18.74 )
c                                            c
1    1  -UDEN   -1    imp:n=1                 c @@@ UFACT = normal 25  1.0 .05
2    0          1    imp:n=0                  c @@@ URAD  = ( UFACT * 8.741 )
                                             c
1      so   URAD                              1    1  -UDEN   -1    imp:n=1
                                             2    0          1    imp:n=0
kcode 10000  1.0   15   115
ksrc  0. 0. 0.                               1      so   URAD
m1    92235 -94.73   92238 -5.27
prdmp 0 0 1 1 0                              kcode 10000  1.0   15   115
                                             ksrc  0. 0. 0.
                                             m1    92235 -94.73   92238 -5.27
                                             prdmp 0 0 1 1 0
```

Figure 2. (left) *mcnp_pstudy* input for varying the fuel density randomly while adjusting radius for constant mass,
(right) *mcnp_pstudy* input for varying fuel density and mass independently

other tests, speedups ranged from roughly 5X-50X, depending on how busy the Linux cluster was at the time of job submittal. The 0.054% uncertainty in Keff for the 1M-history `gdv` problem was reduced to 0.009% for the 50M-history series of `gdv-A` cases.

Figure (2) shows the modified input for the Godiva problem where the fuel density and radius are varied. For problem gdv-D, the fuel density is sampled from a normal distribution with a 5% standard deviation about the nominal density, and the radius is adjusted to preserve the total mass of uranium in

the problem. For problem gdv-E, the fuel radius and density are sampled independently, each from a normal distribution with a standard deviation of 5% about the nominal values.

Table (1) gives the results for the problems illustrated in Figures (1) and (2), as well as for other problems where only the density or only the radius was varied. It can be seen from Problem A that the Monte Carlo uncertainty is small (.0001) and that nearly all of the uncertainties for Problems B-F are due to the statistical sampling for the input parameter values. As noted above, the uncertainties for Problems B-F should not change significantly as more cases are run for each problem. It is interesting to note that the uncertainty for Problem E is significantly smaller that for problem F. In Problem F, both the radius and density are sampled independently; in Problem E only the density is sampled and then the radius is calculated. The correlation between the radius and density serves to reduce the effect of input parameter variation over that of Problem F where 2 independent samples are taken. Also note that the variance for Problem F is approximately equal to the sum of the variances for Problems B and C, as would be expected.

Table 1. Results from varying parameters in the Godiva problem

Problem	Description	K-effective	$\sigma_{K\text{-eff}}$
base	Base case, discard 15 initial cycles, retain 100 cycles with 10K histories/cycle, 1M total histories	0.9970	0.0005
A	Repeat the base problem 50 times, 50M total histories	0.9972	0.0001
B	Vary the fuel density only: sample from a normal distribution with 5% std.dev, 50M total histories	0.9961	0.0061
C	Vary the fuel radius only: sample from a normal distribution with 5% std.dev, 50M total histories	1.0057	0.0051
D	Vary the enrichment only, sample from a normal distribution with 5% std.dev, 50M total histories	0.9890	0.0027
E	Sample the fuel density from a normal distribution with 5% std.dev, and adjust the fuel radius to keep constant fuel mass, 50M total histories	0.9966	0.0042
F	Sample the fuel density from a normal distribution with 5% std.dev, and independently sample the radius from a normal distribution with 5% std.dev, 50M total histories	1.0073	0.0076

Several other examples of applying *mcnp_pstudy* to practical problems illustrate its use in both parameter studies and uncertainty analysis:

- One of these is that of a 55 gallon drum undergoing a non-destructive assay via gamma spectroscopy. By filling the drum with essentially paint can sized items in a hexagonal array (21 close fitting cans total), *mcnp_pstudy* is used to determine what uncertainties can be expected in the assay result as a function of uncertainties in the individual units (paint cans) density, isotopic mix and chemical mix distribution while maintaining a fixed total drum mass and activity. This is done by first perturbing all variables (waste density, elemental distribution and unit activity) on a "can by can" basis under the given constraints by normal deviates of 1%, 5%, 10% and 50%. The actual sensitivity of these total propagated errors are then further understood by the same perturbations but

to only one of the three variables at a time. Finally, the analyses with all three variables perturbed simultaneously are evaluated as a function of source energy by replacing the source energy distribution by discrete photon energies at standard values. The waste matrix and isotopic distribution used as the base case here were taken from that of an average Remote Handled (RH) Transuranic (TRU) waste detailed elsewhere [2].

- The second example takes the same base case drum already described and calculates, for a close fitting external shield, the required thickness to reduce the surface dose rate to 0.5 mrem/hr.

- The third example considers the results of using the base case drum to conduct a "dose to Curie" conversion as would be the case if acceptable knowledge of waste material were being validated based on a single dose rate measurement of the package. This is a method being proposed for meeting waste acceptance criteria for RH TRU waste.

5. Conclusions

A software tool called *mcnp_pstudy* has been developed to automate the setup, execution, and collection of results from a series of MCNP5 Monte Carlo calculations. This tool provides a convenient means of performing parameter studies, total uncertainty analyses, parallel job execution on clusters, stochastic geometry modeling, and other types of calculations where a series of MCNP5 jobs must be performed with varying problem input specifications. The *mcnp_pstudy* software is available at no charge to all MCNP users.

Acknowledgements

REFERENCES

1. 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 (April, 2003).

2. R. B. Hayes, "Dose Rate Prediction Methodology for Remote Handled Transuranic Waste Workers at the Waste Isolation Pilot Plant", *Health Phys.* **83(4)**, 449-455 (2002).