CONF-9505195--4

ASSURANCES ASSOCIATED WITH MONTE CARLO CODE RESULTS

D. F. Hollenbach and L. M. Petrie

Computational Physics and Engineering Division
Oak Ridge National Laboratory†
P.O. Box 2008
Oak Ridge, Tennessee 37831-6370

To be presented at
Nuclear Criticality Technology and Safety Project
Embedded Topical Meeting
Catamaran Hotel
San Diego, California
May 17, 1995

†Managed by Martin Marietta Energy Systems, Inc., under contract DE-AC05-84OR21400 with the U.S. Department of Energy.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

ASSURANCES ASSOCIATED WITH MONTE CARLO CODE RESULTS

D. F. Hollenbach and L. M. Petrie

All Monte Carlo computer codes have an uncertainty associated with the final result. This uncertainty (or standard deviation) is due to the sampling method inherent within the Monte Carlo technique. The basic assumptions required for the final result and uncertainty to be valid are (1) the random numbers used are truly random, (2) there is no correlation between histories, (3) the number of histories used is sufficient to represent the problem, and (4) the entire problem is adequately sampled. The first two assumptions are an integral part of the computer code and the user has minimal control over them. The last two assumptions are strongly dependent on how a problem is set up and the number of histories processed. These are items the user has direct control over. This paper examines six aspects of the KENO Monte Carlo code that affect the above-mentioned four assumptions.

For a random-number generator to work properly, it must have a viable algorithm and a starting seed that will produce long strings of random numbers ($>10^{15}$ numbers) prior to repeating itself. Most random-number generators in use today are based on the linear congruential method. Different machines use different random-number generations. In the workstation version of KENO, the random-number generator combines three independent linear congruential method streams to create a series of random numbers that have a flat distribution. This set of random numbers is used to generate random numbers for other types of distributions. For certain computers and random-number generators the initial seeds have been tested to ensure they produce a long string prior to repeating. If a different sequence of random numbers is desired, for example to run the same problem again, it may be desirable to use one of the random numbers previously generated as the initial random number. This has the effect of simply starting at a different position in the same string of random numbers. If an initial random number is chosen at random, for some random-number generators it is possible the random-number series will either not be sufficiently random or rapidly start repeating itself.

In Monte Carlo codes the uncertainty associated with the final results can be reduced by increasing the number of histories processed. The total number of histories processed is a product of the number of histories per generation and the number of generations run. KENO defaults to 300 histories per generation and 103 generations. This is sufficient to produce good results for a small relatively simple problem. A balance between histories and generations is needed to produce good results. Too few generations and the source distribution may not converge, too few histories per generation and the problem volume may not be completely sampled. Either of these conditions will degrade the quality of the final result and associated standard deviation. The standard deviation generally decreases inversely with respect to the square root of the number histories processed. It is possible to reduce the standard deviation as low as desired by processing more histories. However, beyond a certain point the standard deviation calculated is not representative of the true uncertainty of the final result. This can be especially critical in sensitivity studies where small differences become important.

Machine precision can also have a significant effect on a problem. Single-precision computer codes will not properly process problems with geometry differences of 10^7 or greater in the same coordinate system. On most computers a single precision real number has seven significant figures. If two numbers that differ by greater than 10^7 are added together, the larger number remains unchanged. One needs to also be aware that if two numbers that differ by only 10^5 are added together the smaller number contains only two significant figures regardless of the number of figures it may actually have. Although KENO V.a does its processing in double precision, the geometry data are stored in single precision.

Biasing (or weighting) is used to decrease the CPU time needed to process a problem. Biasing is primarily used on the reflector material surrounding a given fissile assembly or group of fissile assemblies. The most efficient biasing scheme is to weight a history by a function inversely proportional to its importance at that position. KENO has biasing-generated data for concrete, paraffin, water, and graphite based on distances from the fissile source. The biasing data contain six to twenty regions of a specified thickness that go from the lowest biased region next to a fissile region to the highest biased region furthest from the fissile region. If the biasing regions are set up so that upon exiting a high bias region a particle can enter a low bias region or a fissile region without passing through the intermediate bias regions, a large increase in the variance of the results is possible. In KENO, a discontinuity or step change in the standard deviation in the k_{eff} vs generation plot would be an indication of this type of problem.

For the final result to have any value, the source distribution must be converged. For a source to converge, the number of histories per generation must be large enough to represent the system. Given a sufficiently large initial source distribution, a minimum of three and possibly many more generations are usually required before a source will converge to its steady-state result. In some problems the source may never converge but instead will fluctuate between volumes of high reactivity. A good example of this is a spent fuel shipping cask with high burnup, low reactivity, and enrichment in the center, and low burnup, high reactivity, and enrichment at both ends of the fuel assemblies. This problem requires many timed the histories per generation to produce a converged source than is required by the same problem with either a continuous enrichment throughout the fuel assemblies or a cosine distribution enrichment throughout the fuel assembly enrichment. A good method for ensuring the entire problem is properly sampled is to examine the flux densities and fission densities. If a unit containing a large amount of fissile material relative to the other units in the problem has a low flux or fission density, that unit is probably not being properly sampled. This problem may be eliminated by either increasing the number of neutrons per history or explicitly starting histories in that particular unit.

Source convergence can be sped up by specifying an initial source that closely resembles the converged source. In addition to a uniform distribution throughout all fissile material, KENO allows all or part of the initial source to be started as a cosine distribution over a specified volume, in specified units or positions in an array, or at specified positions in the global unit. If a reasonable initial source is not provided, the system may never converge.

As is apparent from the above discussion, simply accepting a result without verifying its quality is a dangerous practice. Simply because a code successfully produces a result does not mean the result is valid. A good understanding of the problem and the code used to solve the problem are essential to producing accurate results. Monte Carlo computer codes should never be used as a black box.