Progress in NUCLEAR SCIENCE and TECHNOLOGY, Vol. 2, pp.866-871 (2011)

ARTICLE

Efficiency Improvement of Local Power Estimation in the General Purpose Monte Carlo Code MCNP

Derk VAN VEEN and J. Eduard HOOGENBOOM*

Delft University of Technology, Mekelweg 15, 2629 JB Delft, The Netherlands

The MCNP general purpose Monte Carlo code was modified to improve the efficiency of detailed power distribution estimation in a reactor core for a large number of tally volumes. The standard MCNP F7 tally for fission energy deposition in a subvolume turns out to take a relatively long CPU time and becomes prohibitively slow when it is applied to all of the large number of fuel pins in a full size reactor, not to mention if it is applied to a number of subvolumes of each fuel pin. The recently introduced FMESH tally uses a Cartesian mesh independent of the material geometry and is therefore much faster, but still generates a lot of overhead. In the proposed modifications the addressing of a non-zero contribution to the deposited energy in a certain subvolume of a fuel pin with regard to the array where all tally contributions are stored is optimized. This turns out to reduce the Monte Carlo simulation time dramatically compared to the F7 tally and still appreciably (about 50 %) compared to the FMESH tally. Another efficiency improvement is the calculation of the potential contribution to the deposited fission energy in a fissile medium as a sum over contributions from all fissile nuclides in the medium in advance of the Monte Carlo simulation and to store this quantity as a function of energy in memory like is done in MCNP for cross section. However, this turns out to result in only a minor improvement in computation time.

KEYWORDS: Monte Carlo, efficiency, MCNP, power estimation

I. Introduction

Although Monte Carlo codes are widely used for reactor core calculations to estimate the effective multiplication factor and energy production in fuel assemblies with sufficient statistical accuracy in an acceptable calculation time, it is currently not feasible to estimate the power distribution in many subvolumes of all fuel pins in a reactor core with sufficient statistical accuracy.[1)] This requires so much CPU time that we cannot only wait for getting access to more and/or faster processors in a computer, but have to develop at the same time major efficiency improvements in our Monte Carlo codes. This may be realized in the neutron simulation procedures, but even more for tallying the energy production in very large numbers of tally volumes.

Improving the efficiency of the neutron history generation will require a complete rewrite of the Monte Carlo code and tailoring to the specific goal, removing the many options offered in a general purpose Monte Carlo code not used for the specific problem. However, improvements in tallying procedures can be realized without restructuring an existing code. Analyzing the energy production tally in MCNP5[2)] it becomes apparent that the procedures used in this code to tally a contribution to the power generation in a specific volume is rather inefficient, partly due to the generality aimed at with this code. In this paper two ways to improve the efficiency of power generation tally are developed and tested.

II. Fission Energy Deposition Estimation in MCNP

The power production P_V in a volume V, or more precisely the fission energy deposition, is given by

$$P_V = \int_V \int_0^\infty \sum_i Q_i N_i(\boldsymbol{r}) \sigma_{f,i}(\boldsymbol{r}, E) \phi(\boldsymbol{r}, E) dE dV \qquad (1)$$

with $\phi(\boldsymbol{r}, E)$ the neutron flux at position $\boldsymbol{r}$ and energy E, $\sigma_{f,i}$ the fission cross section for nuclide i, N_i the nuclide density of nuclide i, and Q_i the recoverable fission energy released in a fission of nuclide i. In MCNP the term fission energy deposition is used as the neutron flux is not sampled in physical units of cm^{-2}s^{-1} but in cm^{-2} per source neutron. Because of the missing time unit the term energy deposition is more correct. Nonetheless, we will often use the term power as it is normally used in nuclear reactor nomenclature.

The fission cross section for each nuclide in Eq. (1) is retrieved at the startup of a job from the MCNP cross section library file and stored in memory at an energy grid covering the full energy range of interest, like it is done for all other cross section needed in the simulation. The fission energy per fissionable nuclide is available from a table included in one of the MCNP modules.

Traditionally MCNP5 offers the so-called F7 tally to estimate the fission energy deposition in a volume using a track-length estimator. At least since the introduction of MCNP5 version 1.30 MCNP5 also offers the general FMESH tally which can be arranged to tally also the fission energy deposition.

*Corresponding author, E-mail: j.e.hoogenboom@tudelft.nl

1. The F7 Fission Energy Deposition Tally

The F7 fission energy deposition tally in MCNP takes an estimate of the flux in volume V in Eq. (1) using a track length estimator and then performs the summation over all nuclides present in the medium of the deposited fission energy. The final result of the estimator is divided by the mass of the volume, giving an outcome in units of MeV/g. At the definition of the F7 tally in the input file for MCNP the user must establish to which geometry cells the tally must be applied. This offers the possibility to restrict the use of the F7 tally to cells containing fuel and not waste time to other cells that won’t give any contribution.

For a reactor core the definition of all fuel pins, cladding and coolant around a fuel pin will be done using repeated structures at different levels. The first level is the repetition in x and y direction of a square unit cell containing a fuel pin with cladding and coolant to fill up a fuel assembly (still allowing certain unit cells with the fuel pin replaced by a guide tube). The second level is the repetition in x and y direction of fuel assemblies to fill up the core (not restricted to a rectangular core). If the energy deposition has to be tallied for different axial volumes of each fuel pin a third level of repeated structure is needed in the z direction.

To include all fuel pins of all fuel assemblies in a core in the F7 tally definition requires to reference each unit cell with its repeated structure indices together with the repeated structure indices of the fuel assembly it belongs to. The different axial volumes to be tallied can be added in a relatively compact way. This amount of input data cannot even be processed by MCNP. Moreover, even a limited number of tally volumes will take a huge amount of CPU time on top of the time necessary for simulating the neutron histories. **Figure 1** shows an example of the CPU time as a function of the number of separate volumes to be tallied. It shows that the CPU time increases roughly linearly with the number of tallies with such a slope that the number of tallies will soon dominate the total CPU time needed for the calculation. Considering that a large reactor core contains about 70,000 fuel pins and that one may want to take 100 different axial volumes per fuel pin, it will not be possible to tally all these volumes with the standard F7 tally.

The F7 tally procedure is programmed in a complicated routine with many tests and nested loops. For the problem type considered here most of the time is spent in looking up whether the particular volume in which a track is registered, is assigned as a tally subvolume. If all tally subvolumes are defined in a single F7 tally definition, the procedure has to check in our case about 7M tally subvolumes for each track. It will complete the do loop after it has found the tally subvolume as in the general case the tally subvolume may show up more than once in the same tally definition. This makes the tally procedure extremely time consuming.

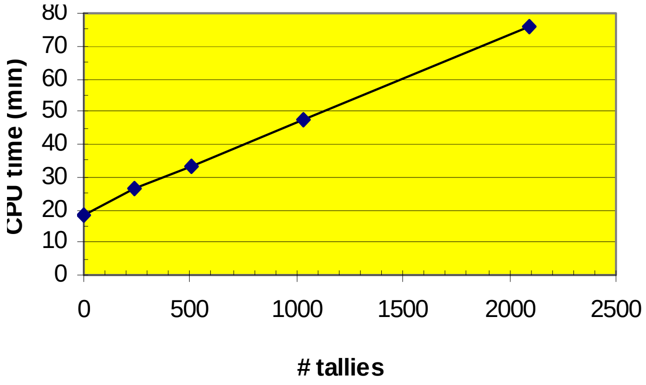

Fig. 1 CPU time as a function of the number of tally volumes

2. The FMESH Tally

The FMESH tally is a general track-length tally which does not record tracks in pre-assigned geometry cells, but uses a mesh superimposed over the problem geometry. The mesh can be defined in Cartesian or cylindrical coordinates. Because of the rectangular geometry of the pin cells in a fuel assembly and of the fuel assemblies in the core we use the Cartesian geometry. The actual mesh can be defined at two levels in each direction, a number of coarse meshes and within each coarse mesh a number of fine meshes. As with other tallies the FMESH track-length estimator can be modified using the so-called FM option in order to get the fission energy deposition. To that end the FM option should indicate that the tally result must be multiplied by the microscopic fission cross section times the fission energy times the nuclide density. If a medium is composed of more than one nuclide the summation over all nuclides are taken, resulting in the fission energy deposition according to Eq. (1).

The FMESH tally is introduced in MCNP to simplify and speedup the tallying process when many tally volumes are involved. This goes at the cost of generality as results from estimates for tracks in different media within a single mesh cell cannot be obtained separately but are all added up. This does not prohibit its use for fission energy deposition as only tracks in the fuel will give a non-zero contribution.

As MCNP5 can simply determine in which mesh cell a certain track is, it can also rather simply determine in which array element of the tally storage array a contribution for a certain mesh cell must be stored. As the starting and end points of a track are determined by a source position and/or collision sites or medium boundary crossing points it is possible that a track extends over two or more mesh cells. The FMESH tally checks whether that is the case and then determines the subtracks limited by mesh boundaries. Although subtracks in different mesh intervals can be in the same material, the FMESH tally will calculate the contribution with the summation over all nuclides in the medium again for each subtrack.

The FMESH tally works much faster than the F7 tally when many tally volumes are involved as the mesh is independent of the complexity of the geometry of the simulated system. Disadvantages are that for the fission energy deposition also all tracks in non-fissile media (coolant, cladding and construction materials) are considered and for each nuclide in such a medium the fission cross section is looked up and found to be zero and multiplied by a fictive fission energy and by the nuclide density. This still leads to a considerable overhead without any contribution to the fission energy deposition. Another disadvantage is that for getting the sum of certain tally results, for instance the energy deposition in a complete fuel pin, in a complete fuel assembly or for the total reactor, separate FMESH tallies must be defined with a proper mesh. This implies that the contribution from a certain track is recalculated for every defined FMESH tally. Although it is possible to add up results from the FMESH tally with the finest mesh for larger volumes like a complete fuel assembly, it is not possible to calculate the correct variance or standard deviation in the sum because of the statistical correlation between the results in the subvolumes.

3. The Lattice Speed Tally Enhancement

MCNP5 offers an option for lattice speed tally enhancement. This option is intended to speed up the tallying of lattices for voxel phantoms, which may consist of large numbers of voxels. The use of this option is only possible for a very restricted class of problems and tallies. As an F7 tally is not allowed for this option it cannot be applied to our problem. Nor is it allowed to use an F4 tally (flux averaged over a volume) with FM option to estimate a reaction rate as will be needed to get the fission energy deposition.

III. Efficiency Improvements

1. Scoring Function

According to Eq. (1) the scoring function contains a summation over all fissionable nuclides in a medium. This summation is done again every time a neutron track in a fuel pin is obtained. Writing Eq. (1) as

$$P_V = \int_V \int_0^\infty h(E)\phi(\vec{r},E)dEdV \tag{2}$$

with the function $h(E)$ given by

$$h(E) = \sum_i Q_i N_i \sigma_{f,i}(E) \tag{3}$$

it will be efficient to store the function $h(E)$ as a function of energy for each different fissile medium instead of doing the summation over nuclides every time again. This requires some additional preprocessing after all cross sections are stored in memory at the start of an MCNP run.

The energy grid for which the h function must be stored is not straight-forward as each nuclide has its own energy grid at which cross sections are stored in memory. Hence, the first step in calculating and storing the h function for a certain (fissile) medium is to take the union of energy grid points of all fissile nuclides in the summation in Eq. (3). If a grid point E_j in the unionized grid is not present in the representation of $\sigma_{f,i}$ for nuclide i, we have to calculate the fission cross section of nuclide i at energy E_j by interpolation in order to obtain $h(E_j)$. Storing $h(E)$ at the unionized grid may require considerable memory space and table lookup time will increase with increasing number of grid points.

We therefore also took into account a few other possibilities[3) instead of using the unionized grid. First it is reasonable to apply a certain thinning to the unionized energy grid, as there may be energy grid points so close together that the h function can be accurately obtained by interpolation between two remaining successive points. We adopted the criterion from Reference 3:

$$\frac{E_{j+1} - E_j}{E_j} < \tau \tag{4}$$

with τ a value that can be set by input to the modified MCNP to delete energy point E_{j+1}. **Table 1** shows the results for using various values of τ with regard to memory space and number of energy grid points. A smaller number of grid points has not only the advantage of less memory usage, but also a more rapid table lookup of the requested function value. With a value of τ=10^{-4} one saves about ¾ of the memory needed while the error is only 0.01 %.

Table 1 Number of energy grid points and memory usage for different levels of grid thinning

Tolerance τ	Number of energy grid points	Memory usage (Mb)
0	796,804	8.40
10^{-6}	737,333	7.83
10^{-5}	449,637	5.14
10^{-4}	165,118	2.27
10^{-3}	83,824	1.24
10^{-2}	70,859	1.06
10^{-1}	69,035	1.03

We tried further methods of speeding up the table lookup like double indexing. In that case apart from an array containing all energies of the grid points, a second, much smaller array is filled with selected energies from the full grid, together with pointers to the index of those energies in the full grid. One can also think of having a grid with uniform spacing, which allows the direct determination of the index in the grid for a given energy. This will not be practical because of the very small energy steps needed in the resolved resonance region of the fission cross section. A variation on this idea is to have three ranges, one say up to 1 eV with constant energy spacing, a second up to 10 keV with constant lethargy spacing for the resonance range and a third above 10 keV with constant energy spacing again. None of all these additional possibilities turned out to be efficient in terms of computation time and memory usage.

To avoid hard coding of options in the modification of MCNP, the possibility offered by MCNP to enter an array of integer and of floating point values by the IDUM and RDUM cards,[2) respectively, in the input file is used. The first element of the IDUM array is used to indicate whether the option with the h function is to be activated and the second array element whether one of the other options as discussed above are to be used. To enter the value of τ for possible thinning of the energy grid for the h function the first array element of the RDUM array is used. If no value for array element RDUM(1) is entered, the default value of zero is in effect and no thinning is applied.

2. Determining the Spatial Tally Bin

The major problem with the F7 tally is to find out in which element of the tally storage array a tally contribution from a certain spatial tally bin must be stored or added. To determine the array element MCNP has to determine in which element of a lattice used to define the repeated structure a track is found. This has to be done at the two or three levels used to define the full core geometry as explained in Section II.1. For the inner level of lattice structures MCNP has to determine whether tallying is requested for the specific geometry cell the track is in. With many tally bins this is a time consuming process.

To avoid excessive lookup and testing we devised a simpler method to determine the tally bin and the storage location in the tally storage array. This requires of course a limitation in the generality of defining tally bins. As the application is the detailed estimation of the power distribution in a large commercial nuclear reactor, we assume that the fuel assemblies contain a regularly ordered array of fuel pins and that the fuel assemblies in the core also form a regularly ordered structure in the *x* and *y* directions.

At both levels the geometry structure from the MCNP input is used to indicate in which lattice element a neutron is at the various lattice levels. In the F7 tally definition in the input to MCNP only the geometry cell numbers containing fuel at the lowest level of the geometry definition are referenced. This excludes any tallying of tracks in non-fissile media.

MCNP stores all scores for separate tally bins in an array TAL together with the cumulative squared value needed after the simulation to calculate the standard deviation. Instead of determining from the tally bin specification in the F7 tally definition in the input file, in the modified F7 tally the TAL array is structured in a strictly regular way in order to determine directly from the lattice indices the array element where to add a tally contribution. This structure also allows for integrated values over each fuel pin, over each fuel assembly and over the whole core. For contributions per fuel pin in an assembly also array elements are reserved for possible unit cell positions with a guide tube and possibly a control rod, which will not give a contribution to the deposited fission energy. This minimizes the time to determine where to store a tally contribution. It will be possible to define an array for all fuel pin positions in a fuel assembly that indicates whether a position contains a fuel pin or a guide tube, if one wants to save memory for storage of the TAL array. As the lattice at the second level that fills up the core extends to the reactor vessel, it contains many reflector elements instead of fuel assemblies. In order not to reserve memory space in the TAL array for positions in the reflector, an array is assigned values of the fuel assembly number and zeroes where reflector elements are present.

Figure 2 shows the ordering of tally bins in the TAL array. Each element in Fig. 2 represents in fact three memory locations, one for the cumulative score during a single history, one for the cumulative score over histories and one for the cumulative squared score.

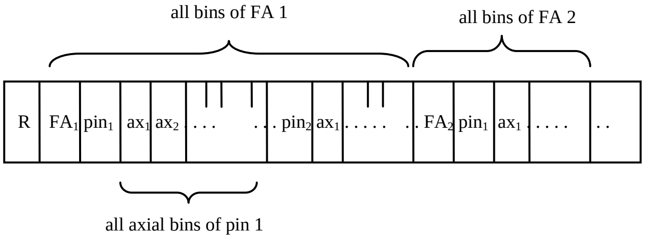

Fig. 2 Order of tally bins in array TAL

To open the possibility of tallying the fission energy in different axial volumes of a fuel pin it is not necessary to define a third level of lattices in the *z* direction along the fuel pins as required by the standard F7 tally. The modified F7 tally treatment uses the second and third RDUM elements in the input to MCNP to determine the lower and upper level of the fuel and the fourth element for the number of axial subvolumes. From these data the code can determine in which axial subvolume the contribution to the tally must be stored. The modified tally determines if a track in the fuel crosses one or more axial subvolume boundaries and calculates the contribution for all subtracks in each subvolume.

The modified F7 tally also adds a tally contribution to the TAL array elements reserved for the total fuel pin, the total assembly and the full core. Together with the sum of squared contributions it allows the proper calculation of the standard deviation for these tallies without any serious overhead.

IV. Demonstration of Results

1. Reactor Model

To demonstrate the correct working of the modifications calculations with MCNP5.1.51 were performed on a realistic full size reactor model extracted from the NEA Data Bank Monte Carlo Performance benchmark.[4)] The reactor core consists of 241 fuel assemblies, each containing 17x17-25 fuel pins. The core is surrounded by water, the downcomer and the reactor vessel. In the axial direction homogenized zones are present for the top and bottom fuel regions, the top and bottom nozzle regions and the upper and lower core plate regions. The fission energy deposition is requested for each fuel pin divided over 100 axial subvolumes, all together making up 6,362,400 tally bins (excluding guide tube positions) + 63,866 bins for the totals over each pin, each fuel assembly and the total reactor core. All fuel assemblies are identical. To introduce an axial asymmetry, the coolant density in the lower half of the core is higher than in the upper half. Although this core shows quadrant symmetry, a full core is modeled, as requested by the benchmark description.

To arrive at a converged fission source distribution 100 cycles of nominally 100,000 neutrons were performed, starting with a spatially uniform source over a cylindrical volume containing the fuel.

2. Validation

To validate the modifications to the F7 tally we compared the results for some specific tally bins with a run with the

Table 2 Comparison of results of the standard and modified F7 tally in MCNP5

Geometry	Standard F7 tally	Modified F7 tally
Reactor	4.80274	4.80274
Fuel assembly (-3,-8)	3.51125 10^{-3}	3.51125 10^{-3}
Fuel Assembly (0,0)	3.81536 $10^{-}3$	3.81536 $10^{-}3$
Fuel pin (0,1)	2.10280 10^{-4}	2.10280 10^{-4}
Fuel pin (-8,-8)	1.72691 10^{-4}	1.72691 10^{-4}
27th axial volume	8.26727 10^{-8}	8.26727 10^{-8}
57th axial volume	1.53643 10^{-6}	1.53643 10^{-6}
Single collision	33.062170444	33.062170438

original F7 tally with a third lattice level to define the axial subvolumes. As we used the same source positions for both calculations and the same random number sequence the results should be the identical, apart from computer word length inaccuracies. **Table 2** shows that this is the case for all selected tallies. The fuel assemblies are indicated with 2 indices in the horizontal plane with (0,0) as the center fuel assembly. In each fuel assembly the fuel pins are also indicated by two indices in the horizontal plane with (0,0) the unit cell in the center of the assembly. The fuel pins selected in the table are in assembly (0,0). The axial volumes are in fuel pin (-8,8) of assembly (0,0). The axial volumes are numbered from 1 at the bottom of the fuel to 100 at the top. The last line in the table gives the contribution to a tally from a single track which is found using a debugger to dig into the code. Its value is not yet normalized and can therefore not be compared with the other values.

3. Efficiency of the *h* Scoring Function

To compare the CPU time when applying the $h(E)$ function from Eq. (3) for calculating the contribution to the fission energy deposition tally instead of the standard F7 tally an MCNP5 run was done of 100 active cycles of nominally 10,000 neutrons, starting from a converged fission source distribution. Grid thinning was applied with a thinning criterion τ=10^{-4}. The run with the standard F7 tally with 924 tally volume bins took 66.63 min CPU time. With the *h* scoring function it took 63.29 min, which is a reduction of about 5 %.

4. Efficiency of Improved Tally Bin Assignment

To demonstrate the effect of the modified F7 tally with improved bin assignment we performed an MCNP5 run with an FMESH tally to obtain the fission energy deposition with a mesh covering each unit cell in the horizontal plane and 100 axial bins, as well as FMESH tallies to obtain the deposition in all fuel pins, in all fuel assemblies and in the total reactor core. This was compared with a run with the modified F7 tally giving the fission energy deposition in the same tally bins. Both calculations were done with 100 cycles of 10^5 neutrons, starting from a converged fission source distribution. Both runs were made on a laptop computer running MCNP5 with one processor. **Table 3** shows the execution times for all calculations. The modified F7 tally uses only about half the time of the FMESH tally, so it gives a clear improvement. Including scoring with the *h* function gives a minor further reduction, which is possibly within the inaccuracy of determining execution times. For comparison we also added the execution time of the standard MCNP5 version without any tally definitions. With the modified F7 tally the additional time used for tallying is relatively small. It also demonstrates that the modified F7 tally works about 20 times as fast as the FMESH tally if we consider only the time for tallying without the time needed for neutron history simulation.

Table 3 Execution time for different options

Tally	Time (min)
No tallies	126.4
Standard F7 tally with 924 bins	655.0
Standard F7 tally with all bins	~4 10^6
FMESH	253.4
Modified F7	132.9
Modified F7 with $h(E)$	131.7

We also compared the time for the standard MCNP F7 tally. To this end we used 924 different volume bins. Assuming a linear relation with the number of bins, we estimated the time needed to do all the bins used with the FMESH and with the modified F7 tally, which results in a prohibitive long execution time.

V. Conclusions and Discussion

The aim of the present work was to improve the efficiency of the Monte Carlo estimation of the detailed power density distribution in a full size reactor with MCNP. It is concluded that the standard MCNP F7 tally for fission energy deposition is totally unsuitable for getting results for large numbers of tally bins. The FMESH tally is more practical, but shows some inefficiencies. To construct a more efficient tally for the fission energy deposition the F7 tally was modified in two ways. By precalculating the scoring function $h(E)$ as a function of energy for each separate fissile medium, the repeated summation over all fissile nuclides in a medium at a certain energy is avoided. Moreover, a faster assignment is introduced of the memory location where a score for a certain tally bin must be stored.

The effects on the execution time for MCNP5 has been tested using a model of a full size reactor core. The introduction of the scoring function $h(E)$ gives only a minor reduction in execution time. Using profiling it became clear that the subroutine for getting the value of the score from the *h* function was a factor 4 faster than the standard MCNP5 routine. However, as the complete tallying process takes about 10 % of the total Monte Carlo calculation time and obtaining the value of the *h* function is only part of the tallying procedure, the total gain remains small.

The efficient assignment of memory locations for the scores, however, turns out to halve the total execution time, which gives a considerable time savings as a Monte Carlo calculation for a full core will take hours to get a reasonable standard deviation in the generated power. This requires proper arrangement of tally bins in the tally score array TAL and use of the indices of the lattices of fuel pins in a fuel assembly and of fuel assemblies in the core. Some further reduction of computing time may be obtained if the non-fissile unit cells with guide tubes instead of a fuel pin are excluded from the tally process using an additional array which should indicate at which lattice positions in a fuel assembly guide tubes are present.

In this example there is only one type of fuel composition used for all fuel assemblies. However, using many different fuel compositions in different fuel assemblies or even in different fuel rods in a fuel assembly will not seriously change the speedup. The efficient assignment procedure of the tally array element does not depend on the fuel material type nor does it make any assumption on repetition of fuel materials. For each track in a fuel cell it will evaluate the fission cross section and the energy deposition per fissionable nuclide again.

The improvement in assignment of tally score storage locations can be generalized for all reactor types that have a regular structure of the fuel pins in the fuel assemblies and a regular structure of the fuel assemblies in the core. Application to a VVER type reactor with hexagonal oriented fuel assemblies seems possible, while in that case the FMESH tally is difficult to apply. For a CANDU type reactor the situation may be more complicated because of the horizontal fuel assemblies and vertical control rods, but considering the regular arrangement of fuel pins in an assembly and of the fuel assemblies in the core, it still seems possible, with some modifications, to apply the current method.

As an illustration of what can be obtained from a full core Monte Carlo calculation **Figure 3** shows the axial power distribution in a specific fuel pin. The axial distribution shows an asymmetry as the coolant densities in the lower and upper half of the core are different. It will be clear that the standard deviation per point is not satisfactory, but it demonstrates that if one can process sufficient neutron histories in an acceptable execution time the final goal of calculating such power distributions in the whole reactor with sufficient statistical accuracy can eventually be reached.

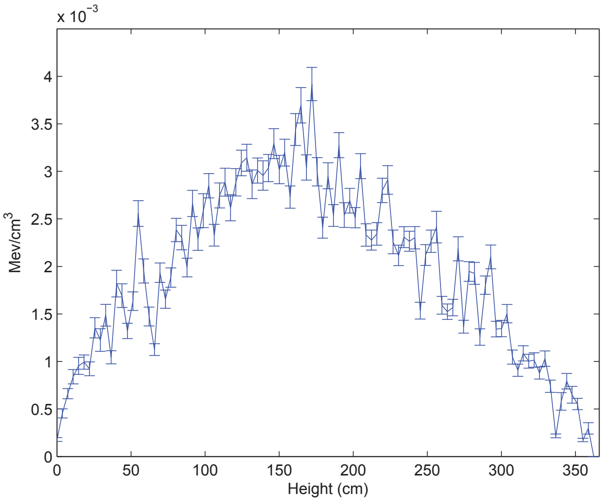

Fig. 3 Axial power distribution in a single fuel pin

References

1) J. E. Hoogenboom, W. R. Martin, “A proposal for a benchmark to monitor the performance of detailed Monte Carlo calculation of power densities in a full size reactor core,” *Proc. Int. Conf. on Mathematics, Computational Methods & Reactor Physics (M&C 2009),* Saratoga Springs, New York, USA, May 3-7 (2009).
2) X-5 Monte Carlo Team, *MCNP – A General Monte Carlo N-particle Transport Code, Version 5*, Los Alamos National Laboratory (LANL) (2003).
3) J. Leppänen, “Two practical methods for unionizing energy grid construction in continuous energy Monte Carlo neutron transport calculation,” *Ann. Nucl. Energy*, **36**, 878-885 (2009).
4) J. E. Hoogenboom, W. R. Martin, B. Petrovic,“ Monte Carlo performance benchmark for detailed power density calculation in a full size reactor core, Benchmark specifications,” Revision 1.1, June 2010, http://www.oecd-nea.org/dbprog/MonteCarloPerformanceBenchmark.htm.