LA-UR-07- 0150

Approved for public release; distribution is unlimited.

Title: The MCNPX Monte Carlo Radiation Transport Code.

Author(s): Laurie S. Waters, Gregg W. McKinney, Joe W. Durkee, Michael L. Fensin, Michael R. James, Russell C. Johns, Denise B. Pelowitz.

Submitted to: Proceedings of the Hadronic Shower Simulation Workshop, September 6-8, 2006, Fermilab, Batavia, Illinois AIP Conference Proceedings.

Los Alamos
NATIONAL LABORATORY
EST 1943

The MCNPX Monte Carlo Radiation Transport Code

Laurie S. Waters, Gregg W. McKinney, Joe W. Durkee, Michael L. Fensin, John S. Hendricks, Michael R. James, Russell C. Johns, Denise B. Pelowitz

Los Alamos National Laboratory, MS K575, Los Alamos, New Mexico 87545

Abstract. MCNPX (Monte Carlo N-Particle eXtended) is a general-purpose Monte Carlo radiation transport code with three-dimensional geometry and continuous-energy transport of 34 particles and light ions. It contains flexible source and tally options, interactive graphics, and support for both sequential and multi-processing computer platforms. MCNPX is based on MCNP4B, and has been upgraded to most MCNP5 capabilities. MCNP is a highly stable code tracking neutrons, photons and electrons, and using evaluated nuclear data libraries for low-energy interaction probabilities. MCNPX has extended this base to a comprehensive set of particles and light ions, with heavy ion transport in development. Models have been included to calculate interaction probabilities when libraries are not available. Recent additions focus on the time evolution of residual nuclei decay, allowing calculation of transmutation and delayed particle emission. MCNPX is now a code of great dynamic range, and the excellent neutronics capabilities allow new opportunities to simulate devices of interest to experimental particle physics; particularly calorimetry. This paper describes the capabilities of the current MCNPX version 2.6.C, and also discusses ongoing code development.

Keywords: MCNPX, MCNP, Monte Carlo, Radiation Transport, Particle Transport, Neutronics
PACS: 24.10.Lx

INTRODUCTION

MCNPX [1] development began in 1994 with funding from the APT (Accelerator Production of Tritium) project. APT used a 1 GeV proton beam on a tungsten spallation target to produce a high neutron flux. The neutrons were moderated in a surrounding lead blanket, and captured in He-3 gas to produce Tritium. Due to the critical emphasis on neutronics in this project, the primary simulation tool was the LAHET (Los Alamos High Energy Transport) Monte Carlo code system [2]. LAHET is a set of codes allowing transport of 18 particles, and includes a high energy package which used the Bertini [3], ISABEL [4,5] and FLUKA [6] models for primary interactions. Dresner evaporation [7], a Fermi Breakup model [8], and both the ORNL [9] and Rutherford-Appelton [10] fission models were included to model residual de-excitation. LAHET was one of the first codes to include a multistage pre-equilibrium exciton model [11] as an intermediate stage between the Intranuclear Cascade (INC) and evaporation phases of a nuclear interaction. Neturons produced by LAHET below 20 MeV were written to a file for later transport by the MCNP code.

The MCNPX project was initiated to simplify the process of using the LAHET package, to extend the particle base, to enable the addition of new interaction models for high energy physics, and to provide upgrades to the capabilities of the latest MCNP code. Rather than including low energy neutronics physics in LAHET, it was decided to expand the capabilities of MCNP4B to include more particles, and physics models to calculate interaction probabilities when evaluated data libraries were not available. Initially, the standard LAHET physics packages described above were included, later development added the CEM and LAQGSM models [12], the INCL4 INC model [13], and the ABLA evaporation/fission code [14]. The APT project also sponsored the development of entirely new evaluated data libraries for photonuclear and proton interactions, and also expanded the upper energy range of neutron evaluations for 42 isotopes to 150 MeV. APT also funded improvements in the CINDER'90 burnup code, which was recently incorporated directly into MCNPX and allows simulation of transmutation and delayed particle production.

Since its inception, MCNPX has focused on the needs of the intermediate energy community, here taken to mean incident energies up to a few GeV. This involved a strong focus on accurate INC, evaporation and fission models, and a close coupling with traditional neutronics methods. This has enabled applications for a wide variety of projects, such as spallation target and accelerator shielding design, cosmochemistry, medical physics and Homeland Security applications. The initial addition of certain FLUKA physics raised the energy limits above traditional INC limitations, and the more recent inclusion of LAQGSM has greatly improved code capabilities to model very high energy interactions. The code now ranges from TeV energies to thermal neutrons. The inclusion of reaction time-evolution and the ever expanding set of allowed particles has given MCNPX applicability for experimental particle physics applications, especially in modeling very low energy experimental backgrounds.

THE CODE STRUCTURE

The following sections will describe the code availability and infrastructure, the geometry and source definition capabilities. Problems in MCNPX are specified with well-defined lines in input files, and no user coding is necessary to run a problem.

MCNPX Availability and Basic Structure

MCNPX is available from RISCC, and to a limited number of beta testers, as explained on the official code web site, http://mcnpx.lanl.gov. The RSICC release is bundled with MCNP5, the evaluated data libraries needed to run the code, and several subsidiary libraries needed for the physics models in MCNPX. A few additional codes are included primarily for formatting of data libraries, and to convert tally information into formats readable by outside analysis packages. Although it is too early to predict a release date, MCNPX and MCNP are in the process of formally merging into one code package. MCNPX also comes with an extensive User's Guide [1], and regular updates on code features are given in release notes which can be examined on the web site. Five-day beginning, intermediate and advanced classes are held on a regular basis, with the latest schedule available from the web site.

MCNPX is written in Fortran 90, with a few C++ routines primarily involving the graphics capabilities. The code can be downloaded as a binary for PC, LINUX, MAC, Sun Solaris and IBM AIX platforms. The source code is also available and can be prepared through a special autoconf-generated configure script distributed with the code. This allows the user to set a number of parameters such as the specific compiler, location of libraries, multiprocessing options, and enabling of code features controlled by specific keywords. Actual compiling is done by a MAKE utility, either GNU MAKE, or a version supplied by the system vendor.

MCNPX supports distributed memory multiprocessing for the entire range of all particles. Parallel Virtual Machine (PVM) [15] and Message Passing Interface (MPI) [16] can be used to run the code in parallel. Fault tolerance and load balancing are available, and multiprocessing can be done across a network of heterogeneous platforms. Threading may be used for problems run in the tabular data region only. Many recent MCNPX applications have been done on Beowulf clusters ranging in size from a few to hundreds of nodes.

The code is controlled by a user-written line-formatted input file, with sections describing cell geometry, surface specification, and physics/tallying options. Two interactive GUIs are available for problem setup: MORITZ [17], and VISED [18] which is part of the official RSICC release. CAD geometry input is available through the GUIs and an embedded CAD capability is currently under development.

Geometry and Source Capability

MCNPX uses the MCNP surface-based geometry, where defined surfaces are combined into geometrical cells. An interactive geometry viewer is included in the code for debugging purposes. Available surfaces include planes, spheres, cylinders, cones, ellipsoid/hyperboloid/ paraboloid and an elliptical or circular torus. Each surface has a 'sense' defined by the surface normal, and cells are defined by combinations of surfaces, where a point within a cell is defined relative to the various surface senses. Surfaces are combined with Boolean operators; intersection, union and complement. Thus it takes 6 defined planes to describe a box. Recently this process has been simplified by the addition of Macrobody surfaces, which conveniently combine the surfaces needed for common shapes into a single surface description. Ten objects are available, including the box, rectangular parallelepiped, sphere, right circular cylinder, right hexagonal prism, right elliptical cylinder, truncated right angle cone, ellipsoid, wedge, and an arbitrary polyhedron. The Macrobody surfaces are used exactly like regular surfaces in the cell description. Due to variable dimensioning in the F90 code, there is in principle no limit to the number of surfaces or cells that may be defined in a problem.

Surfaces can be designated as 'reflecting', whereby any particle hitting the surface will be specularly (mirror) reflected. This aids in setting up a finite boundary problem which mimics an infinite volume. Surfaces can also be designated as "white", where particles hitting the boundary are reflected with a cosine distribution relative to the surface normal. Periodic boundary conditions may also be applied to pairs of planes to simulate a simple infinite lattice. More complex lattices are described below.

Every cell must be filled with a material at a fixed density. The materials are designated as a combination of elements, either by weight or atomic fraction. The code's "Mix and Match" capability allows the user to specify if interactions will be performed through reference to evaluated nuclear data, directly computed with a model, or some combination of the two, on an isotope by isotope basis. Materials are assumed to be solid, unless a gas phase is specifically identified, and material temperature may also be specified for the free-gas treatment of thermal neutrons. It is also possible to specify conducting materials for electron-transport problems.

Both cells and surfaces can be rotated and translated by applying a transformation description consisting of an offset, and/or a rotation matrix. Entire cells may be duplicated elsewhere in the problem, with changes allowed to cell material, material density, orientation, and other values related to variance reduction. The entire physical area of a problem must be specified out to infinity, which defines a 'universe'. Several different 'universes' may be defined in a problem, and cells of finite volume may be 'filled' with different universes. This allows an individual cell to be filled with widely varying geometries, much as a window in a wall shows a view of part of the outside world. Some or all cells in a universe may themselves be filled with other universes. These filled cells may also be replicated into rectangular or hexagonal lattices. The "fully specified fill" capability of the code allows each element of a lattice to be 'filled' with a universe of a different material. Figure 1 shows an example of a CT scan of a head specified with lattice geometry in MCNPX.

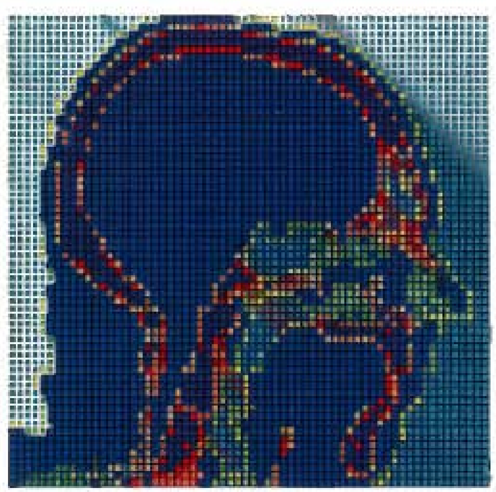

FIGURE 1. CT scan of a head using the fully-specified lattice fill capability of MCNPX.

Sources in MCNPX are designed by identifying the particle type, energy, time, position and direction. Sources can fill a cell, start from a surface, or form a geometry independent of the actual cell geometry. More than one particle type may be specified in a single source. It is often possible to specify one variable as a function of another variable, for example, energy can be a function of particle type. Energies can also be specified with preset functions, such as Maxwell or Watt fission energy spectra. Or, individual energies may be listed along with associated probabilities, either individually or as a histogram. Energy spectra can also be read in from an external file, or the results of a calculation may be written to a file and read in and sampled for further transport in a different problem. MCNPX can currently automatically generate spontaneous fission neutrons of appropriate multiplicity and energy if the user includes an actinide in the material description. Work is now in progress to allow the user to designate a radioactive material as a source, and the code will read in the decay gammas from a library. If no specifications are made, the default source is an isotropic 14 MeV neutron located at the problem origin.

TRANSPORT AND INTERACTION PHYSICS

MCNPX currently tracks 30 different particles and 4 light ions. Tracking is done to a user-settable lower kinetic energy cutoff, and particles will decay with their standard halflives. The physics needed to run a simulation will be discussed in three parts. First, we review the MCNPX capabilities for charged particle transport; ionization, multiple scattering and energy straggling. Second, the tabular based physics options are discussed. Lastly, we briefly review the inline model physics used when tabular based options are not available or desired.

Charged Particle Transport

For heavy charged particles, the fundamental Bethe-Bloch formalism has been enhanced to include values and interpolation procedures recommended in ICRU Report 37 [19], bringing the model into closer ICRU compliance. The density effect correction now uses the parameterization of Sternheimer and Peierls [20]. For high-energy protons and other light charged projectiles, the approximate SPAR model [21] has been replaced with a full implementation of the maximum kinetic energy transfer. For intermediate energies, the shell corrections to the stopping power have been adapted from Janni [22]. Finally, a continuous transition in the stopping power between the ranges 1.31 MeV/AMU (Atomic Mass Unit) for the high-energy model, and 5.24 MeV/AMU (the low energy SPAR model) is achieved with a linear interpolation between the two models. Small angle Coulomb scattering uses the Rossi-Griesen algorithm. In the original theory, both angular deflections and small spatial displacements were accounted for. MCNPX does not yet accommodate transverse displacements in charged-particle subsets, therefore we only use the part of the theory that addresses the angular deflections. This is a slight effect and test cases have found negligible effects on the results. Energy straggling uses Vavilov logic, recently improved [23].

Evaluated Data Libraries

Nine classes of evaluated data libraries are available, with the most familiar being continuous-energy neutron, photonuclear, proton, neutron $S(\alpha,\beta)$ thermal data, photoatomic (up to 100 GeV), and electron interaction data (up to 1 GeV). Photoatomic and electron data are atomic in nature, and therefore depend only on Z. All others contain isotopic data. The $S(\alpha,\beta)$ data takes into account scattering in specific molecular structures, and substitutes for very low energy cross sections. These must be evaluated for materials, not isotopes, therefore only commonly used materials have yet been calculated.

Photon interaction libraries include Thompson, Compton, Photoelectric and Pair Production cross sections. The electron libraries contain ionization and bremsstrahlung data. The neutron and photonuclear libraries contain total, elastic, (n,xn), fission, (n,γ) cross sections, and often heating numbers. All library data can be plotted directly in MCNPX, for individual isotopes, or for models. Many collections of evaluated libraries, such as ENDF, ENDL, JEF have been formatted for use in the MCNP-type codes, primarily through the NJOY code. In the future, libraries may become available for very high energy interactions, but presently the community has not yet reached agreement on the physics to sufficient accuracy to warrant this effort.

The traditional process of sampling data libraries sometimes makes individual particle tracking difficult. Upon interaction, all channels are sampled, and the correlation of one interaction to the next is lost. Energy is conserved on average, but not on an individual track basis. Codes like PoliMi [24] are available which restore the correlation.

Physics Models

Most of the physics models included in MCNPX (listed in the Introduction) are discussed in papers in this conference, and will not be detailed here. The general sequence of a calculation note starts with an INC model. After a time or energy cutoff (depending on the model), a pre-equilibrium phase is entered. A decision is then made to either fission or de-excite. De-excitation can follow an evaporation or Fermi-breakup model. Below the neutron emission threshold, evaporation proceeds by photon emission. The recent addition of the CINDER'90 transmutation code makes it possible to follow the time evolution of residuals, alone, or in interaction in the presence of a neutron fluence.

Figure 2 shows the results of Benchmark Problem 7 for this workshop. Energy deposition is calculated for an incident 50 GeV proton beam on a 10 cm long, 1 cm radius cylindrical tungsten target. The two MCNPX curves illustrate a problem sometimes encountered by MCNP users, and shows that methodologies which work well at very low energies do not often extrapolate to very high energies. The "Thick Target Bremsstrahlung model" is often used to speed up calculations. Electrons are generated, but their energies are deposited locally. Bremsstrahlung photons produced by the non-transported electrons are fully tracked. This works well at energies where produced photons do not have much energy, but is inappropriate for high-energy physics applications. The upper MCNPX curve shows local electron energy deposition. Full tracking of electrons gives a more consistent answer with other codes.

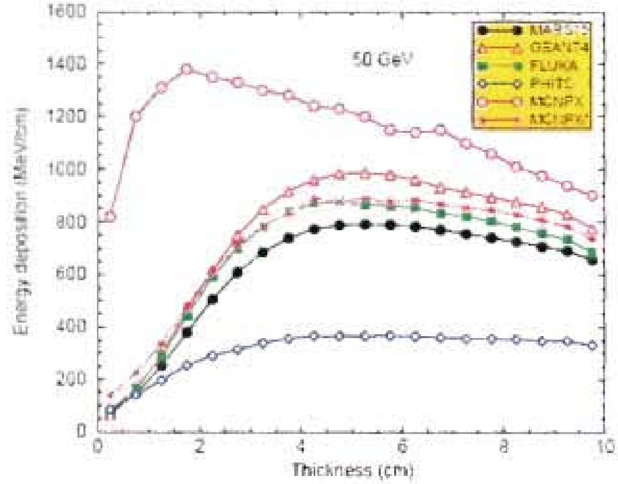

FIGURE 2. Effect of local energy deposition of electron energies..

TALLYING AND ANSWER CONVERGENCE

In the MCNP series of codes, answers are known as 'tallies'. The user specifies what tally is desired for what particles and for what region of the geometry, all with simply constructed entries in the problem input file. With any radiation transport code, three questions must be addressed; what is calculated, what is the accuracy of the answer, and how fast can the calculation be done. MCNPX has well-tested pre-defined tally choices for current, fluence and energy deposition, eliminating the need for the user to define these quantities from first principles. Rather than rely solely on variance, MCNPX has ten statistical tests that can be used to assess tally convergence. A large variety of variance reduction techniques are available in the code, and the skilled user can substantially reduce the amount of time needed for tally convergence, even with traditionally difficult problems. In discussing tallies and convergence, it is important to note that every particle in MCNPX carries a weight with it at all times. If the weight is 1.0, then every particle is counted as one particle. Weight can change for several reasons, primarily variance reduction. Depending on the technique used, particles can be split or multiplied, with their weights adjusted so that the answer can be proven to be a fair game with the mean value theorem.

MCNPX has four types of tallies, Track Length, Surface, Collision and Next Event estimators. Most of these can be calculated as a function of energy, angle and time. Track length refers to flux/fluence tallies in a volume. Flux is fluence per second, and it is up to the user to normalize the answer per unit time if desired. Flux can easily be weighted by quantities such as number density, cross sections, fluence-to-dose conversion factors, and heating factors in order to calculate physical observables.

Many of these factors are contained in the evaluated data libraries. The MCNPX estimator for volume flux in a cell is the sum of all particle weights times their tracklengths in the cell, and divided by the cell volume.

It is an easy exercise to show that flux across a surface is the sum of particle weights divided by both surface area and the absolute value of the cosine of the particle direction with respect to the surface normal. MCNPX does have the capability to designate angular bins for surface flux, however the normal uses of volume flux do not need such a distinction. The particle will interact no matter what direction it is going. Another example of a Surface tally is current; the simple sum of particle weights crossing a surface, which can be divided into angular bins. A more complex type of Surface tally is the "pulse height" tally, used for calculating the energy deposited in a cell. The energy of particles leaving a volume is subtracted from the energy of the particle entering the volume, all for a single source event. This quantity can be folded in with detector response functions. Recent work in MCNPX has focused on making pulse height tallies work with variance reduction, and allowing coincidence and anticoincidence with energy deposition in other detectors.

Collision estimators are primarily used to calculate energy deposition, and proceed in two different ways. If a library is available, heating factors are folded into a flux. In the model regime, separate components of energy deposition (ionization, residual recoil, local energy deposition for non-tracked particles, energies of particles falling below tracking cutoffs) are added together. This differs from the pulse-height tally in that the calculation is done on a particle-by-particle basis, not event-by-event.

Next-Event estimators, in MCNPX are represented by 'point' and 'ring' detectors, allow the user to estimate fluxes at specific points, or on a ring if the problem has cylindrical symmetry. In evaluated data libraries, angular distributions of scattered particles are tabulated. Thus, the probability that such a scattering heads in the direction of interest is known, and in a flux calculation, the particle weight is multiplied by this probability. The weight is also divided by the square of the distance to the point, and multiplied by an exponential attenuation factor representing the attenuation lengths of intervening materials. Both direct and scattered contributions to the tally are calculated. Arrays of point detectors are used to calculate either pinhole or transmitted radiography images. Point and ring detectors do not currently work with charged particles, or in the model region where angular distribution are not tabulated.

Tallies can be viewed with a graphics routine directly in the code, either after completion of a calculation, or while the problem proceeds. The "mesh tally" is also available. This is a method for viewing any of the tally types over a large physical area. A user-defined 3-dimensional rectangular, cylindrical or spherical mesh is laid over the existing geometry, and particles tracked both through the mesh and geometry simultaneously. A quantity such as flux is calculated by summing weighted particle tracks through a mesh cell, and dividing by the volume of that cell. Mesh tallies can be viewed with the tally plotter, overlaid on the problem geometry, or reformatted for viewing with an outside plotter.

Five separate quantities are tracked to assess tally convergence. These include the tally mean, relative error, variance of the variance, and "Figure of Merit" (defined as variance divided by the number of histories), and the slope n in $1/x^n$ of the largest history scores of the probability density function . The code looks at the value of each quantity, and performs certain tests as the problem progresses. For example, the relative error should be less than 5%, decrease as the problem progresses, with decrease rate going as the inverse of the square root of the number of source particles run. The Figure of Merit should be constant, with random fluctuations. The user will have an efficiently running problem if FOM is large. It has been shown that a slope parameter greater or equal to 3.0 will not allow the problem to converge. The statistical check results can be printed for each tally, but it is up to the user to determine if the result is consistent with a converged answer. One common problem is inadequate sampling of the source distribution, which can cause large jumps in these quantities as the problem proceeds; the answer rarely settles down.

Variance reduction options in MCNPX are extensive, and basically include four types: Truncation, population control, modified sampling, and partially-deterministic methods. Point and ring detectors are examples of partially-deterministic methods. Another is the DXTRAN sphere, which is similar to the point detector, except that particles biased toward a sphere around the point of interest, and can be further tracked within the sphere. This allows the calculation of detector responses in small, distant cells. Truncation refers to limiting the problem by stopping the calculation at pre-determined limits, such as a lower energy cutoff in particle tracking, or a time cutoff. Modified sampling methods include options such as biasing source particles in certain angular directions, energies or times of interest.

Population control relies on particle splitting and Russian roulette to control the number of samples taken in various regions of phase space. One use of this is the 'weight window' methodology. A weight cutoff is specified, and particles above this weight are split, below the weight are Russian Rouletted. A user-defined mesh can be placed over the geometry, and weight window limits written to each cell in the mesh. In 'automated weight window' calculations, the weight windows are automatically adjusted by the code over successive runs to properly converge a particular tally located in a difficult area of the geometry.

SUMMARY

Current developments in MCNPX are largely tuned to the needs of the threat reduction, and space science communities. We are adding tallying capabilities which focus on determining the origin of signals – specific reactions, or from particular residual nuclei. The code's ability to calculate delayed particle signals is under aggressive development. Photonuclear physics improvement is a continuing program, while both muon capture and resonant fluorescence physics will be added in the coming year. We are also improving our source capabilities – automatic generation of gamma lines from specific isotopes; as well as individual photon line data from delayed particle emission. Several approaches to correlated particle production are under investigation. Heavy ion transport is close to completion. Above all, MCNPX maintains a high standard of quality assurance, and welcomes feedback from all users.

REFERENCES

1. D. B. Pelowitz, ed., "MCNPX User's Manual Version 2.5.0", Los Alamos National Laboratory, LA-CP-05-0369, April 2005.
2. R. E. Prael and H. Lichtenstein, "User Guide to LCS: The LAHET Code System", Los Alamos National Laboratory, LA-UR-89-3014, Revised (September 15, 1989)
3. H. W. Bertini, "Intranuclear-cascade Calculations of the Secondary Nucleon Spectra fro Nucleon-Nucleus Interactions in the Energy Range 340 to 2900 MeV and Comparisons with Experiment", *Phys. Rev.* **188** 1711 (1969).
4. Y. Yariv and Z. Fraenkel, *Phys. Rev. C* **20** 2227 (1979).
5. Y. Yariv and Z. Fraenkel, *Phys. Rev. C* **24** 488 (1981).
6. G. Collazuol, A. Ferrari, A. Guglielmi and P. R. Sala, "Hadronic Models and Experimental Data for the Neutrino Beam Production", *Nuclear Instruments & Methods* **A449** 609-613 (2000).
7. L. Dresner, "EVAP – A Fortran Program for Calculating the Evaporation of Various Particles from Excited Compound Nuclei", ORNL-TM-196, Oak Ridge National Laboratory (April 1962)
8. D. J. Brenner, R. E. Prael, J. F. Dicello, and M. Zaider, "Improved Calculations of Energy Deposition from Fast Neutrons", in *Proceedings Fourth Symposium on Neutron Dosimetry*, EUR-7448, Munich-Neuherberg (1981).
9. J. Barish et. al., "HETFIS High-Energy Nucleon-Meson Transport Code with Fission", ORNL/TM-7882, Oak Ridge national Laboratory (July 1981).
10. F. Atchinson, "Spallation and Fission in Heavy Metal Nuclei under Medium Energy Proton Bombardment", in *Targets for Neutron Beam Spallation Sources*, Jül-Conf-34, Kernforschungsanlage Jülich GmbH (January 1980).
11. R. E. Prael and Michael Bozoian, "Adaptation of the Multistage Preequilibrium Model for the Monte Carlo Method (I)", LA-UR-88-3238, Los Alamos National Laboratory (September 1988).
12. S. G. Mashnik, A. J. Sierk, K. K. Gudima and M. I. Baznat, "CEM03 and LAQGSM03 – New modeling tools for nuclear applications", EPS Euroconference XIX Nuclear Physics Divisional Conference, *Journal of Physics: Conference Series 41*(2006) pp 340-351.
13. A. Boulard, J. Cugnon, S. Leray and C. Volant, "Intranuclear cascade model for a comprehensive description of spallation reaction data", *Phys. Rev. C* **66**, 044615-1 to 044615-28 (2002).
14. A. R. Junghans, M. de Jong, H.-G. Clerk, A. V. Ignatyuk, G. A. Kudyaev, K.-H Schmidt, "Projectile-fragment yields as a probe for the collective enhancement in nuclear level density", *Nuclear Physics A* **629** 635-655 (1998).
15. CSM, Oak Ridge national Laboratory, "Parallel Virtual Machine (PVM)", http://www.csm.ornl.gov/pvm/pvm_home.html.
16. MCS, Argonne National Laboratory, "Message Passing Interface (MPI) Standard", http://www-unix.mcs.anl.gov/mpi/.
17. http://www. whiterockscience.com/moritz.html
18. http://www/mcnpvised.com
19. International Commission on Radiatio Units and Measurements (ICRU), "Stopping Powers for Electrons and Positrons," Report 37, October 1984.
20. R. M. Sternheimer and R. F. Peierls, *Phys. Rev. B3, no. 11* (June 1, 1971), pp 3681.
21. T. W. Armstrong and K. C. Chandler, "Spar, A Fortran Program for Computing Stopping Powers and Ranges for Muons, Charged Pions, and Heavy Ions", ORNL-4869, (May, 1973).
22. J. F. Janni, "Proton Range-Energy Tables, 1keV-10GeV," *Atomic Data and Nuclear Data Tables 27*, (1982) 2/3
23. R. E. Prael, "Proposed Modification to the Charged Hadron Tracking Algorithm in MCNPX", Los Alamos Research Note X-5-RN (U), LA-UR-00-4027, (August 23, 2000).
24. S. A. Pozzi, E. Padovani, and M. Marseguerra, "MCNP-POLIMI: A Monte Carlo Code for Correlation Measurements", *Nuclear Instruments and Methods in Physics Research A, Vol 513, Issue 3* (2003), pp 550-558.