ANS MC2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method • Nashville, TN • April 19-23, 2015, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2015)

penORNL: A PARALLEL MONTE CARLO PHOTON AND ELECTRON TRANSPORT PACKAGE USING PENELOPE

Kursat B. Bekar, Thomas M. Miller, Bruce W. Patton, and Charles F. Weber*
Oak Ridge National Laboratory
1 Bethel Valley Rd., Oak Ridge, TN 37831, U.S.A
bekarkb@ornl.gov; millertm@ornl.gov; pattonbw@ornl.gov; webercw@ornl.gov

ABSTRACT

The parallel Monte Carlo photon and electron transport code package penORNL was developed at Oak Ridge National Laboratory to enable advanced scanning electron microscope (SEM) simulations on high performance computing systems. This paper discusses the implementations, capabilities and parallel performance of the new code package. penORNL uses PENELOPE for its physics calculations and provides all available PENELOPE features to the users, as well as some new features including source definitions specifically developed for SEM simulations, a pulse-height tally capability for detailed simulations of gamma and x-ray detectors, and a modified interaction forcing mechanism to enable accurate energy deposition calculations. The parallel performance of penORNL was extensively tested with several model problems, and very good linear parallel scaling was observed with up to 512 processors. penORNL, along with its new features, will be available for SEM simulations upon completion of the new pulse-height tally implementation.

Key Words: Monte Carlo, PENELOPE, parallel computing, SEM

1 INTRODUCTION

penORNL is a new parallel framework developed for performing Monte Carlo (MC) simulations of photon and electron transport on small to mid-size computing clusters, especially for scanning electron microscope (SEM) applications. It uses the PENELOPE code system [1] for physics calculations, and it provides several source definitions specifically designed for SEM simulations, a pulse-height tally capability for detailed simulations of gamma and x-ray detectors, and a modified interaction-forcing mechanism for accurate energy deposition calculations. Although the goal of this development work is to improve the efficiency of the MC calculations for SEM processes, the final penORNL product will be available for other applications that involve photon and electron transport calculations. This paper discusses the development, implementation, capabilities, and parallel performance of penORNL for SEM simulations.

---

2 MOTIVATION FOR PENORNL DEVELOPMENT

A finely-tuned electron beam of low-to-moderate energy (typically 5 – 35 keV) strikes a sample material and backscattered electrons enable the user to obtain high-resolution images and detailed topography of a sample surface. The electron interactions also generate x-rays from fluorescence characteristic of the elements present, and the elemental composition of the samples can be determined by examining the x-ray spectra. This process is called Electron Probe Microanalysis (EPMA). By focusing an electron beam to a diameter of a few nanometers, the x-rays can determine elemental composition (qualitative analysis) with a spatial resolution well below one micrometer. Using the proper electron beam energy, most of the elements on the periodic table can be reliably detected with this methodology. Similarly, the relative abundances of a sample can be measured reasonably accurately if the sample satisfies stringent criteria (flat, highly polished, upright relative to the beam, chemically uniform) for comparative analysis with known, carefully prepared standards. In fact, this type of quantitative analysis has not been realistic for many real-world applications because the samples do not satisfy those criteria. This work is designed to overcome these restrictions to provide both qualitative and quantitative analysis without using standards.

Our approach is to use MC simulations as models of samples to rapidly determine sample properties, including both qualitative and quantitative chemical composition. Several MC codes for this purpose were evaluated, and PENELOPE/PENEPMA [1,2] was identified as the best starting point for SEM simulations [3,4] based on the following:

- The extensive data and physics models in the code ensure that all photon peaks can be reproduced so as to accurately describe all details of actual spectra.
- Overall code performance in simulation time and accuracy is superior compared to other codes, although MCNP6 [5] was also excellent in this regard.
- The source code is available and can be easily modified.

PENELOPE, a general-purpose MC simulation code for electron and photon transport, is widely used for radiotherapy and nuclear medicine, dosimetry and radiation metrology, electron microscopy, and detector simulations. PENELOPE implements an elaborate mixed scheme [6] for high energy electron transport processes that use accurate cross sections for elastic scattering and accounts for energy-straggling effects. Photon interactions are simulated in chronological succession, allowing the calculation of x-ray fluorescence in complex geometries. PENELOPE also has a variance reduction mechanism based on interaction forcing for both electrons and photons, which greatly improves the MC statistics if the interactions are rare. PENEPMA [2] is a dedicated main program of PENELOPE designed to simulate electron microanalysis experiments with x-ray detectors.

In spite of the advantages listed above, the PENELOPE code still has some deficiencies for the purpose of detailed SEM simulations and requires additional development. One of the major drawbacks of PENELOPE/PENEPMA is that it does not have parallel execution capability, which is beneficial to reduce the simulation time, especially for complex models, by running the code on many CPUs concurrently. Similarly, a pulse-height tally capability is not available—an important feature that is necessary to model x-ray detectors realistically rather than empirically. To address these issues and improve the PENELOPE utilization for SEM simulations, a new parallel modular code package, penORNL, has been developed.

3 PENORNL DEVELOPMENT

Monte Carlo is considered to be the most accurate method for conducting radiation transport calculations; however, MC has a number of disadvantages, such as numerical and computational inefficiencies that increase the computer run time and create uncertainties in the reliability of the results. Researchers have been working to address these issues for decades by developing several variance reduction techniques and parallel MC algorithms.

The penORNL code was developed to address similar issues in PENELOPE/PENEPMA Monte Carlo simulations for SEM. The new code was designed in a modular structure for easy implementation of new features. It was developed in Fortran2003 using object-oriented programming features. It uses a distributed memory parallel programming model via MPI [7].

In this section, penORNL parallel implementation, reorganization of PENELOPE and PENEPMA codes, and new penORNL features are discussed.

3.1 Parallel Implementation

One approach for parallelizing the MC algorithm is *domain replication*, which is a simple form of parallelism and usually causes lower communication overhead compared to more complex approaches. In this approach, the same problem is replicated on a number of processors, and a MC simulation is performed on each replica by running some portion of the problem. This approach can be implemented for both shared and distributed memory systems using the corresponding parallel programming model.

The simplest form of this type of parallelism, the "master-slave approach," was used for penORNL development. In this strategy, one process is allocated as the *master process* to set up the problem, establish communication among the processes, broadcast data for domain construction on each process, gather computed data from the other processes, and control the other processes for a successful parallel execution. The remaining processes are allocated as *slave processes* to perform only particle transport calculations on the given problem domain. The slave processes usually communicate only with the master process, and not with the other slave processes. In this approach, the master process can also perform particle transport calculations to improve the process utilization by decreasing the idleness of the master process.

A data communication library was developed to support all MPI features, to provide data serialization for complex data structures (e.g., user-defined derived-types), and to implement timing routines that evaluate the serial/parallel code performance. Methods in this library provide a transparent communication interface among the modules in the penORNL framework. A communication interface was designed using this data communication library to provide data transfer between master and slave processes, as depicted in Fig.1. This interface involves the following three communication steps:

1. **Collective communications:** Master process broadcasts global data. Problem data are transferred from master to all slaves before starting particle transport.
2. **Peer-to-peer communication:** Master sends process-specific data to each slave. Individual data are transferred from master to a specific slave during the particle transport if necessary.
3. **Collective communication of results:** Master receives results (locally stored tallied quantities) from all slaves using MPI's global reduction functions. Frequent tally updates

and the number of tallies used in the problem increase the collective communication overhead for the given problem. penORNL allows users to adjust this frequency by setting either a time interval or the batch size.

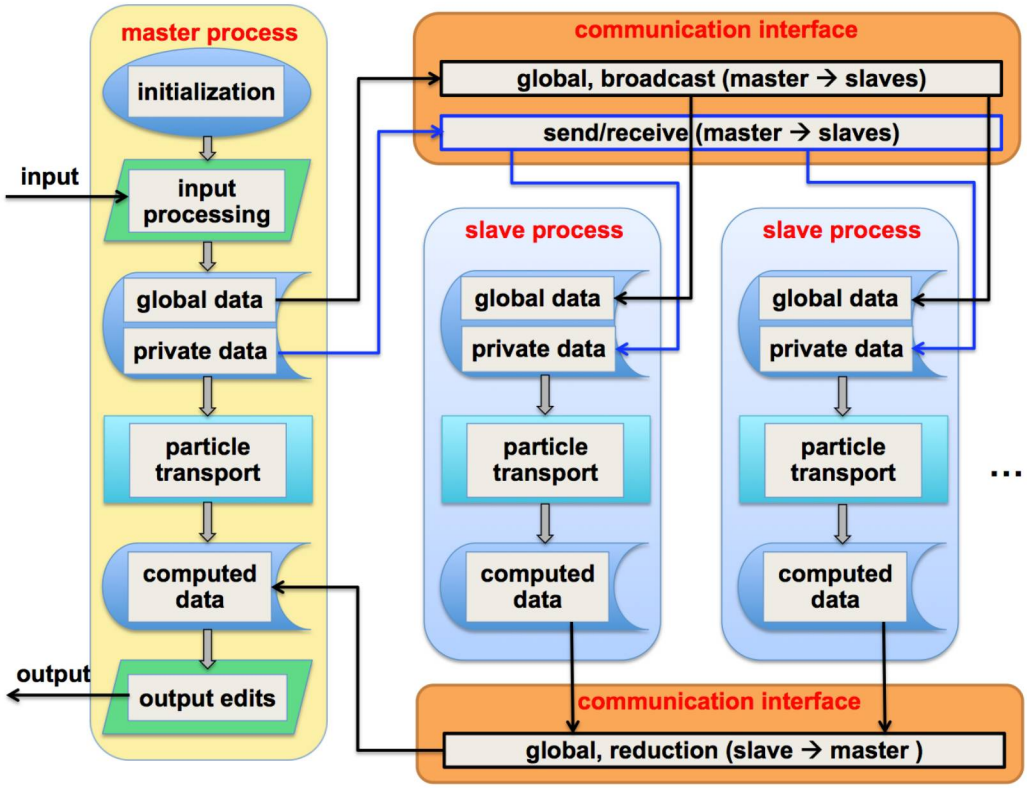

Figure 1. Master-slave organization in penORNL parallel implementation.

In this implementation, code parallel performance is limited by the serial code sections in the master process (input processing and output edits/post-processing), and by the communication overhead among the modules. The more time the code takes in communication and serial section computation, the lower the parallel performance attained by the code. However, MC simulation of electron transport requires a significant amount of time compared to the time required for input and output processing and for communications. Furthermore, the required data at each communication level are appended in a memory buffer which is transferred from one processor to another. In this way, multiple communication operations with small chunk-size data are reduced to improve the efficiency of the parallelism. As a result, very good scaling was observed for most of the SEM simulations. Note that the communication interface is inactive for the serial calculation, where only the master process performs the entire simulation.

Another advantage of this framework is to enable data and method extensions either in the serial or the parallel code sections. In addition, an application-programming interface (API)—which specifies a software component in terms of its operations, their inputs and outputs and underlying types—was designed for the particle transport module to enable a straightforward integration of the particle transport code and its utility functions to the framework.

3.2 Reorganizing PENELOPE/PENEPMA Modules

To integrate the PENELOPE/PENEPMA code into the new parallel framework, all functions and subroutines in the code system were grouped in terms of their functionality into the following four different packages:

1. **penLib:** PENELOPE routines for MC particle transport and interaction-sampling,
2. **penGeom:** PENELOPE routines for geometry,
3. **penVared:** PENELOPE routines for variance reduction, and
4. **penRita:** PENELOPE routines for random number and random-number sampling methods for single-variate discrete and continuous probability distributions.

The PENELOPE and PENEPMA main drivers were decomposed into several subroutines and moved into these packages depending on the relevance of the contents of each package. Code sections for tally and source calculations were also separated from these packages and from the main programs to enhance the modularity of each package. In this way, the contents of each package can be easily replaced or updated, depending on the requirements. Tally functions and other particle counters were redesigned within the penORNL framework. They will be defined in a separate package before the final version of the code is released.

The random number generator plays an important role in MC particle transport simulations to achieve reliable numerical results. It is not desired to have identical particle tracks for the different histories on each computational node. Therefore, a random number generator was developed with the history-based random number seed generation feature. In this way, it is guaranteed that in a simulation, each particle starts its random walk with the same random number when the code runs with a different number of MPI tasks. In other words, the new implementation enables identical random walks for a particular history during the transport process when running the code with different number of processors (i.e., exactly reproducible results with different numbers of processors). The original random number functions in the penRita package were replaced with an improved random number generator for more reliable simulations. penORNL not only uses random numbers for particle tracking in the parallel section, but it also might use them for other calculations in the serial section. Therefore, another random number sequence is necessary for the calculations in the serial section. To restart penORNL properly from the previously saved restart data file, the random number seed is saved to the restart data. It is not necessary to save the random number seeds in the parallel section to the restart data file because they can be regenerated with the history numbers.

After modifications for each package were finalized, they were integrated to the penORNL parallel framework and used for SEM simulations.

3.3 Features of penORNL

Unique features of penORNL are described in detail in the following subsections.

3.3.1 Source module

A general source module was developed to support all available PENELOPE and PENEPMA source definitions, as well as four new source definitions specifically designed for SEM

simulations. This module provides the methods to read, process, and check the source definitions specified in the input file, and it constructs the source arrays/objects for the transport calculation. The following four source definitions for both electrons and photons were introduced:

1. **single/multiple point source(s):** generates isotropic source particles at each location specified in the user input,
2. **single/multiple pencil beam source(s):** generates source particles that are mono-directional at locations specified by the user (i.e., a focused beam with some distortion),
3. **grid beam source:** generates multiple point or pencil beams on a user-specified grid, and
4. **plane source:** generates point or pencil beam sources at locations randomly selected on a specified plane.

These new source definitions enable penORNL to simulate the raster-scanning process that occurs in a typical SEM process, in which a single x-ray spectrum results from successive beam interactions at all points on a Cartesian grid.

3.3.2 Tally module

A general tally module was developed within the penORNL framework, and existing tally calculation routines from PENELOPE code system and PENEPMA implementation were moved to this new tally module. A new interface was designed between the tally module and the particle transport module to enable required tally data such as the track length between the two collision sites, amount of energy deposited after a collision process, particle information, etc. The new tally module design also enhances the modularity of the new parallel framework.

3.3.3 Modified interaction forcing

PENELOPE offers interaction-forcing mechanisms to improve the occurrence probability of a rare event in electron and photon transport simulations. Using such a variance reduction technique can reduce the uncertainties greatly for the same number of particle histories.

This methodology is described in the current PENELOPE manual [1] as follows:

- The interaction probability of a process for a given particle in a specific region is artificially increased using a user-specified forcing parameter, (F)
- The mean free path (λ) of a particle is scaled using F, ($\lambda' = \lambda/F$)
- Weights of secondary particles (W) are scaled with F if they exist, ($W' = W/F$)
- The parent state is modified with the probability ($1/F$)

Although this technique significantly reduces the uncertainties for most simulations, it has a deficiency: this technique violates energy conservation, because an incoming particle may transfer most of its energy to the secondary particles in a specific interaction, but it may keep its state without any modification. Therefore, the sum of energies deposited along a track differs from the energy lost by the projectile; simulations with this technique yield energy deposition spectra that are biased. This deficiency limits the use of PENELOPE in SEM simulations because scoring false-energy deposition results in inaccurate energy depositions and pulse-height tallies. Therefore, an effort is under way to correct this problem. This modification will be tested with several sample problems, including models for gamma spectroscopy and energy dispersive x-ray spectroscopy (EDS), when the pulse-height tally implementation is completed.

3.3.4 Pulse-height tally implementation

Increasingly, SEM machines are equipped with silicon drift-based detectors (SDD) to measure x-rays. An SDD is a pulse-height detector that counts the number of pulses of different amounts of energy deposited in the detector crystal. In order to simulate this type of detector more realistically, a detector pulse-height tally capability was integrated into the new tally module of penORNL. In this implementation, the pulse-height tally bins represent different amounts of energy deposited, as in a pulse-height detector. Currently the pulse-height tally feature in penORNL is only operational without variance reduction mechanisms like interaction forcing. Different approaches are being considered to use this feature with an interaction forcing mechanism.

4 PARALLEL CODE PERFORMANCE

The code integrity and parallel reliability of the penORNL code were tested on a computational cluster with several MPI processes. In the testing stage, the code suite was run both in serial and parallel modes using the test problems in the regression suit. Results from these calculations indicate that identical results were obtained from both serial and parallel modes. The same test problems were run with PENELOPE and PENEPMA, and their results were compared to those from penORNL simulations. Because the penORNL code uses a different random number generator than PENELOPE/PENEPMA, its results are not numerically identical to those of PENELOPE/PENEPMA but they are statistically equivalent. All these testing results show that penORNL successfully performs MC coupled electron-photon transport using PENELOPE/PENEPMA physics, and it is applicable to SEM simulations.

The performance of the MPI implementations in penORNL was determined through a series of timing comparisons with test problems. In this section, speedup plots show the *theoretical speedup,* which is proportional to the number of parallel tasks in MPI parallelism. The theoretical speedup is calculated as if the code was parallelized 100% (no serial section) and has no communication overhead. Therefore, the theoretical speedup is equal to n, the number of parallel tasks.

Figure 2 shows a performance measure for the penORNL code, which very nearly achieves the theoretical maximum with up to 512 MPI tasks. This case involves a sample problem with simple geometry in which a 25 MeV electron beam impinges on a 0.2 cm-thick tungsten sample. In this case, simulations were performed with $N = 10^6$ source particles using various numbers of MPI tasks (n). Inclusion of excessive secondary-particle-generation biasing in the tungsten block slows down the entire transport process. As a result, the parallel section time increases significantly relative to the serial section and communication time, and parallel speedup approaches the theoretical value (indicated by the red line) with a parallel efficiency of 98–100% for this problem.

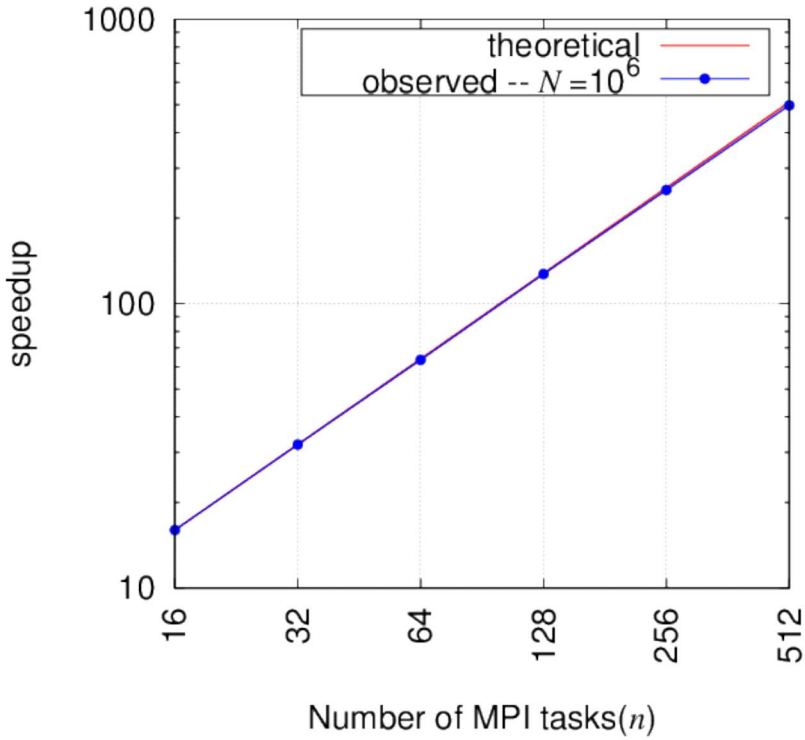

Figure 2. Parallel speedup for penORNL calculations for a given sample problem (0.2 cm thick tungsten slab with a 25 MeV electron beam source).

Another sample problem was designed to demonstrate the code parallel performance for a typical electron microprobe analysis (EPMA). In this study, a simple geometry was designed as a thin aluminum substrate with a 1 μm diameter uranium sphere embedded in it. A 25-keV electron beam was used, and several tallies were set to calculate the x-ray emission spectra from these materials. Simulations were performed without any variance reduction mechanisms with two different numbers of source particles: $N = 10^8$ and 10^9. These values are typical for this type of analysis to obtain detailed x-ray emission spectra with reasonably acceptable uncertainties. This sample problem is relatively simpler than the first sample problem in terms of transport processes; therefore, the wall-clock time per source particle is shorter compared to the first sample problem.

Parallel speedups for the calculations with the second sample problem with two different numbers of source particles are presented in Fig. 3. Results show that the observed parallel speedup approaches the theoretical maximum (denoted by the red line) with $N = 10^9$ source particles, but it does not scale as well for the case with $N = 10^8$ source particles. This is because the computational load on each MPI process is smaller than the case with $N = 10^9$ source particles. The computational load on each MPI process almost linearly decreases with the number of MPI tasks, and similarly, the communication time among the MPI processes increases with the number of MPI tasks. Therefore, the wall-clock time for the parallel section (transport process) approaches the wall-clock time of the code's serial section plus the communication time. As a result, communication overhead limits the parallel performance, and the code parallel performance degrades as the number of MPI processes declines. It should be noted that this is similar to the behavior of other parallel Monte Carlo particle transport codes.

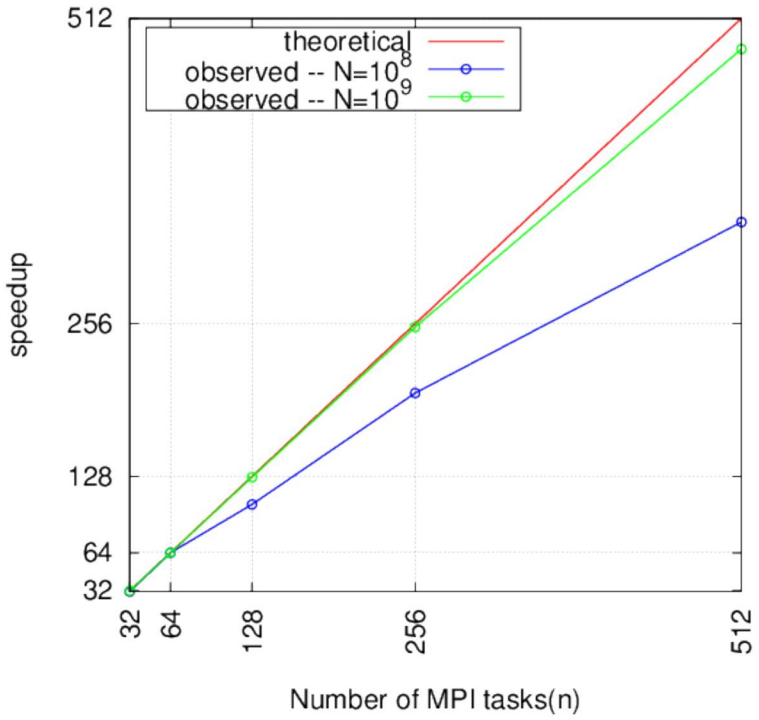

Figure 3. Parallel speed up for the penORNL calculations for a given sample problem (Al substrate with a micron size U sphere, with a 25 keV electron beam source).

5 CONCLUSIONS

penORNL was designed to efficiently perform parallel Monte Carlo simulations of photon and electron transport on distributed memory systems. The main framework of the code was developed in Fortran 2003 using object-oriented programming models to facilitate the integration of PENELOPE and other modules with minimal effort and time. Along with the existing available PENELOPE and PENEPMA features, new features were added to the code system to address advanced SEM simulations for qualitative and quantitative analysis. Timing results for testing the code parallel performance exhibit near perfect parallel efficiency. PENELOPE's variance reduction mechanism was also modified to conserve energy and to support accurate energy deposition calculations and pulse-height tally calculations. The final product will be provided with all available PENELOPE features and with new capabilities not only for SEM calculations, but also for the other applications (radiotherapy and nuclear medicine, microdosimetry and radiation metrology, detector response, x-ray generators, etc.) that involve photon and electron transport calculations.

6 ACKNOWLEDGMENTS

This work was performed with funds provided by the US DOE Office of Defense Nuclear Nonproliferation Research and Development. The authors wish to thank Francesc Salvat, primary developer of PENELOPE and PENEPMA, for his helpful feedback.

7 REFERENCES

1. F. Salvat, J.M. Fernández-Varea, J. Sempau, "PENELOPE 2011: A code System for Monte Carlo Simulation of Electron and Photon Transport," *OECD/NEA Data Bank*, Issy-les-Moulineaux, France (2011). Available in PDF format from http://www.nea.fr/lists/penelope.html.
2. X. Llovet and F. Salvat, "PENEPMA, a Monte Carlo code for the simulation of x-ray emission spectra using PENELOPE", Code Manual (2008).
3. T. M. Miller, C. F. Weber, and B. W. Patton, "Monte Carlo Simulations of X-ray Spectra from Scanning Electron Microscopes," *ORNL/LTR-2013/467* (2013).
4. T. M. Miller, B. W. Patton, and C. F. Weber, "Simulation of Electron Probe Microanalysis for the Purposes of Automated Material Identification—Initial Evaluation of Available Monte Carlo Tools," *Trans. Amer. Nucl. Soc.* **110**(1), 497 (2014).
5. J. T. Goorley, et al., "Initial MCNP6 Release Overview - MCNP6 version 1.0," LA-UR-13-22934 (2013).
6. F. Salvat, "Class II Algorithm for Charged-Particle Transport Simulation," *18th Topical Meeting of the Radiation Protection and Shielding Division of ANS*, Knoxville, TN, USA, September 14-18 (2014).
7. E. Gabriel et al., "Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation," *Proceedings, 11th European PVM/MPI Users' Group Meeting*, 97–104, Budapest (2004).