

---

ARTICLE

---

## Massively Parallel Monte Carlo: Experiences Running Nuclear Simulations on a Large Condor Cluster

James TICKNER<sup>1,\*</sup>, Greg HITCHEN<sup>2</sup>, Joel O'DWYER<sup>1</sup>, Greg ROACH<sup>1</sup> and Josef UHER<sup>1</sup>

<sup>1</sup>CSIRO Process Science and Engineering, Sydney, NSW 2234, Australia

<sup>2</sup>CSIRO Information Management and Technology, Kensington, WA 8555, Australia

The trivially-parallel nature of Monte Carlo (MC) simulations make them ideally suited for running on a distributed, heterogeneous computing environment. We report on the setup and operation of a large, cycle-harvesting Condor computer cluster, used to run MC simulations of nuclear instruments ('jobs') on approximately 4,500 desktop PCs. Successful operation must balance the competing goals of maximizing the availability of machines for running jobs whilst minimizing the impact on users' PC performance. This requires classification of jobs according to anticipated run-time and priority and careful optimization of the parameters used to control job allocation to host machines. To maximize use of a large Condor cluster, we have created a powerful suite of tools to handle job submission and analysis, as the manual creation, submission and evaluation of large numbers (hundred to thousands) of jobs would be too arduous. We describe some of the key aspects of this suite, which has been interfaced to the well-known MCNP and EGSnrc nuclear codes and our in-house PHOTON optical MC code. We report on our practical experiences of operating our Condor cluster and present examples of several large-scale instrument design problems that have been solved using this tool.

**KEYWORDS:** *high-performance computing, Monte Carlo, particle transport, grid computing, optimization*

### I. Introduction

Monte Carlo (MC) radiation techniques are widely used for modeling the transport and interactions of X-rays, gamma-rays, neutrons and other high energy particles. A wide variety of computer codes are available for simulating different types and energies of radiation.<sup>1-4)</sup> The main advantages of the MC method are its ability to simulate microscopic physical interactions according to known process cross-sections and its capability of representing arbitrarily detailed geometric models. The main drawbacks of the technique are twofold: the effort to set up detailed simulations of complex apparatus and the computer run time required to achieve acceptable statistical simulation accuracy, with errors generally reducing only as the square root of the execution time.

Our group develops X-ray, gamma-ray and neutron-based instruments to address imaging and composition analysis problems in the minerals and security industries. We use MC simulation techniques to conduct rapid prototyping and feasibility studies and to design and optimize new analysis instruments. These studies are characterized by the need to create models for large numbers, often hundreds or thousands, of analyser design permutations and rapidly evaluate their performance.

In this paper we report on a suite of tools that we have developed to meet these needs. An interactive graphical interface facilitates setting up of all aspects of a MC simulation, including specification of the analyser geometry,

materials, radiation source, tally quantities and variance reduction techniques. Permutations of the basic design of an instrument can be created either manually by the user or automatically through a simple scripting mechanism. The optimal design for a particular application can be identified by manual inspection or automatically using a genetic-algorithm based optimization tool.

To provide the computational resources needed to run large number of MC simulations, we make use of the Condor High Throughput Computing<sup>5)</sup> software. Running on approximately 4,500 of our organization's machines distributed across Australia, the Condor software uses cycle-harvesting techniques to allocate waiting jobs to idle desktop computers. We report on our experiences using this large virtual cluster and the tools we have developed to automate the processes of job submission, monitoring and retrieval.

### II. XPERT: the Job Specification Interface

Creating a MC model of a nuclear instrument is a time-consuming and exacting process. Typically, the user must describe the geometry, materials, radiation source, tallies and variance reduction techniques to be used either by writing a dedicated program that calls functions provided by the MC code's API<sup>3,4)</sup> or by creating a text-format input file that is parsed by the MC code.<sup>1,2)</sup> Either approach requires considerable expertise on the part of the user and makes exploring variations on an instrument design both tedious and error-prone.

We have developed a graphical interface, XPERT, which

---

\*Corresponding author; Email: james.tickner@csiro.au

allows MC models to be created, modified and run far more quickly and by relatively inexperienced users. We use this interface with the well known MCNP<sup>1)</sup> and EGSnrc<sup>3)</sup> codes for neutron/gamma and electron/gamma transport respectively and with an in-house optical photon transport code PHOTON.<sup>6)</sup>

### 1. Geometry Definition and Validation

Specifying the 3-dimensional geometry is one of the most time-consuming aspects of creating a model of a nuclear instrument. The model must divide space into a number of discrete, homogeneous regions containing different materials, corresponding to the different physical components of the instrument.

We have developed a constraint-based approach<sup>7)</sup> to geometry specification where model components are described using constructive solid geometry (CSG) operations on a range of simple primitives such as rectangular cuboids, cylinders and spheres. Primitives are positioned and dimensioned by placing constraints on points of a rectangular cuboid volume – the bounding box – which is just large enough to contain the primitive. Changes can be made to the constraints at any time and these propagate automatically through the entire geometry model. This approach allows objects to be created in a natural way and facilitates subsequent variations and refinements whilst preserving model consistency.

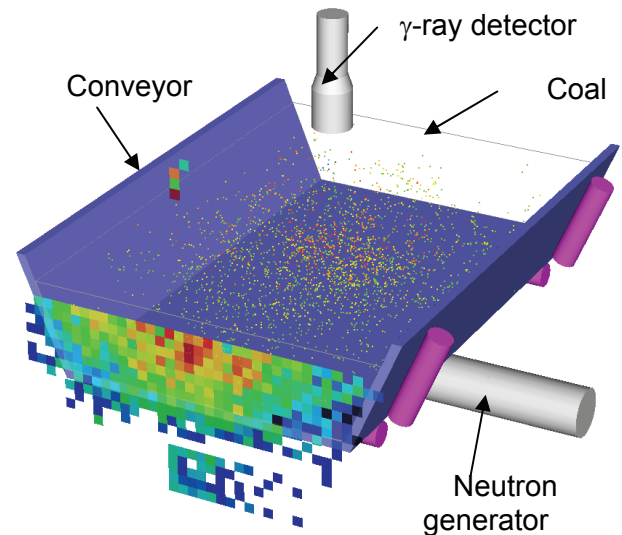
The geometry model can be viewed in whole or in part in 2- and 3-dimensions and tools are provided for automatically detecting specification errors. The underlying MC physics codes can be called interactively, with particle tracks and tally results overlaid onto the geometry model. **Figure 1** illustrates a simple model of an analyser.

Coal passes along a conveyor belt. A 14 MeV neutron generator irradiates the coal and neutron-induced gamma-rays are measured by a BGO detector positioned above the belt. The points show where individual carbon gamma-rays are produced and are colored according to the strength of the signal they produce in the detector. The colored squares at the end of the belt section show the integrated contributions along the belt and provide an indication of the spatial uniformity of the carbon analysis.

### 2. Source, Material and Tally Libraries

A comprehensive library of radioisotope radiation sources is included and an interface is provided to add new isotopes by downloading the necessary data from the ENDSF database.<sup>8)</sup> Source modeling includes gamma-ray cascade coincidences, X-ray and electron or positron emission.<sup>9)</sup> Functions are provided to calculate continuum X-ray emission spectra for X-ray tube or high-energy LINAC based sources using the analytical models of Refs. 10) and 11). The user can also manually define arbitrary sources if required and add these to the library. Once a source has been selected, its specification (energy distribution, radiation type, and angular distribution) are automatically converted to a form suitable for use by MCNP, EGSnrc or PHOTON.

A library of predefined materials is also provided. The



**Fig. 1** Rendering of simple model of a neutron-gamma coal-analyser. See text for details

user can define new materials by atomic composition or chemical formula and add these to the library. Isotopic compositions are automatically calculated based on known isotope ratios for the elements unless overridden by the user.

A wide variety of tally types are provided for the user. In addition to the basic flux, current and energy-deposition tallies, more physically relevant outputs can be calculated. These include detector responses (estimated using a combination of a particle-counting tally and a detector response function), spatially segmented flux and energy deposition tallies useful for radiographic simulations and human-equivalent dose rate and reaction-rate estimators. Again, the code-specific implementation details are hidden from the user by the XPERT interface.

### 3. Instrument Permutations

Designing and optimizing a nuclear instrument involves exploring the space of design permutations. This space might include the geometric configuration of the source, detectors and material being measured; the operating parameters of an X-ray tube or choice of radioisotope source and the composition of structural and shielding materials. Creating and maintaining separate files describing each design permutation becomes a significant problem.

To overcome this, we provide two mechanisms for representing and managing design permutations of an instrument model. The first allows the user to specify a 'base' design together with an integer number of *variations*. The mapping of variation number to the change in the instrument design is completely arbitrary and controlled by a simple scripting language interface. The interface allows the user to create and assign values to 'variables' (text strings) that can appear wherever a number is required, for example when specifying the position or size of a geometric region. The user can also directly create new regions and modify material or source parameters based on the variation number. The

variation approach is best suited to exploring a set of specific design choices.

The second approach allows the user to specify one or more *optimization parameters*. These are floating point variables that may take any value over the range 0-1. Again, the mapping from parameter values to the change in instrument design that they produce is completely arbitrary and specified using the same scripting mechanism used for the variations approach. The optimization parameter approach is best used for general multi-parameter optimization problems.

#### 4. Optimization Strategies

Identifying the optimal instrument design for a given problem requires an optimization strategy. For simpler problems – those with few parameter choices or where the parameters can be independently chosen – a user-driven approach may be sufficient. The user creates and runs a number of design permutations and evaluates their performance according to relevant criteria. For example, if designing a trace-element analyzer, the performance criteria might be the statistical error on the concentration measurement for the element of interest, which can be estimated from the signal and background count rates in the modeled detector response.

For more complex problems, we have developed a genetic algorithm based optimizer.<sup>12)</sup> This algorithm searches the space of design possibilities parameterized by the user. Random permutations are created by choosing random values for the optimization parameters. MC input files for these permutations are created and run. Tally outputs are returned and a figure-of-merit calculated from these using a script supplied by the user. Features from the best performing designs are recombined and reevaluated. Eventually, the algorithm converges to best performing designs.

### III. Processing on a Large Condor Cluster

Rapidly evaluating large numbers of instrument designs requires significant computational resources. The trivially parallel nature of MC simulation, together with its relatively modest memory and file-size requirements, makes it well suited for a heterogeneous, distributed computing cluster. The Condor high-throughput computing software<sup>3)</sup> allows unused computing capacity in an organization to be harnessed to run CPU-intensive applications. Users submit jobs from their PC using software tools provided by Condor. Jobs are copied directly to a host machine which then runs them; once complete, output is copied back to the original submitting computer. Job queuing and user/host matching is performed by a central server, which provides powerful options for scheduling, prioritizing and monitoring jobs.

#### 1. Cluster Configuration

The CSIRO Condor cluster comprises approximately 4,500 desktop machines distributed across 37 sites in our organization. These nodes are split into 5 regional Condor "pools" that are connected together via a Condor mechanism

known as "flocking". The Central Managers for each pool are virtual machines running SLES10 Linux that are hosted on ESX servers in each region.

A couple of minor differences to the default Condor configuration settings have been implemented for CSIRO. All communication "interval" settings have been reduced from 5 minutes to 30 seconds to improve system responsiveness. In addition, two classes of jobs have been defined. One is for normal longer running instances. The system has been configured so that these can only start execution outside of normal working hours (overnight or weekends). The other class consists of short running jobs of less than 30 minutes. These can start at any time but will target machines that have been idle the longest. In both cases the jobs will only start on nodes that show no user activity. In addition a job will be instantly evicted if any user activity is detected (mouse, keyboard, or CPU use). In the standard configuration, a user can run a maximum of 200 jobs concurrently from a single Windows PC.

All installation of the Condor software onto Windows nodes is performed automatically at logon time and checks for updated binaries and configuration files are scheduled daily.

A CondorView server collects, collates and displays machine and user statistics on a weekly and monthly basis. For the past 2 years from August 2008 to July 2010 the CSIRO Condor cluster has averaged 25,000 jobs (3 CPU compute years) per month.

#### 2. Interface with XPERT

Normally, working with Condor requires the use of command-line instructions from a DOS or UNIX prompt window. For example, submitting jobs requires the user to prepare a script file detailing the executable to be used, listing input and output files and specifying other job requirements. Whilst relatively straightforward, shielding users from unnecessary low-level system commands is an important part of the XPERT design philosophy. We have developed a simple graphical interface that allows users to submit jobs, monitor their progress and retrieve output files (see Fig. 2).

To submit a job, the user merely selects one or more MC problem description files on their computer. The XPERT program locates the necessary executable and material data files, prepares the submit script and enters the jobs into the Condor queuing system. The user can elect to submit multiple instances of each job, in which case the random number generator is reseeded for each instance to ensure statistically independent runs.

Monitoring is carried out using a combination of local file interrogation – looking for the output files that are copied back from host machines when jobs finish – and queue monitoring commands provided by Condor. Output files from multiple runs of the same problem are automatically combined, with tally results and errors recalculated for the integrated run. This makes it easy for the user to split long jobs into many parts, utilizing the trivially parallel nature of the MC approach.

ID	Job	Type	Status	Submitted	Run time
39.000	cuactiv3-0001.egs	EGS	DK	Aug 04 13:53	
39.001	cuactiv3-0002.egs	EGS	DK	Aug 04 13:53	
39.005	cuactiv3-0006.egs	EGS	DK	Aug 04 13:53	
39.006	cuactiv3-0007.egs	EGS	DK	Aug 04 13:53	
39.007	cuactiv3-0008.egs	EGS	DK	Aug 04 13:53	
39.008	cuactiv3-0009.egs	EGS	DK	Aug 04 13:53	
39.009	cuactiv3-0010.egs	EGS	DK	Aug 04 13:53	
39.011	cuactiv3-0012.egs	EGS	DK	Aug 04 13:53	
39.015	cuactiv3-0016.egs	EGS	DK	Aug 04 13:53	
39.018	cuactiv3-0019.egs	EGS	DK	Aug 04 13:53	
39.019	cuactiv3-0020.egs	EGS	DK	Aug 04 13:53	
39.002	cuactiv3-0003.egs	EGS	Running	Aug 04 13:53	00:01:20
39.003	cuactiv3-0004.egs	EGS	Running	Aug 04 13:53	00:01:20
39.004	cuactiv3-0005.egs	EGS	Running	Aug 04 13:53	00:01:20
39.010	cuactiv3-0011.egs	EGS	Running	Aug 04 13:53	00:01:20
39.012	cuactiv3-0013.egs	EGS	Running	Aug 04 13:53	00:01:20
39.013	cuactiv3-0014.egs	EGS	Running	Aug 04 13:53	00:01:19
39.014	cuactiv3-0015.egs	EGS	Running	Aug 04 13:53	00:01:19
39.016	cuactiv3-0017.egs	EGS	Running	Aug 04 13:53	00:01:19
39.017	cuactiv3-0018.egs	EGS	Running	Aug 04 13:53	00:01:20

Status: 20 jobs [11 completed, 9 running]

Fig. 2 Condor job status monitoring tool in XPERT

### 3. Operating Experiences

Making effective use of a cycle harvesting high-throughput computing cluster means balancing the needs of users and the owners of the host computers, the majority of whom do not use Condor. Avoiding any perceptible degradation in the performance of the host machines is critical.

By default, our Condor cluster is configured to only run jobs outside of office hours. Jobs requiring up to 12 hours can be run over night and longer jobs up to 60 hours can be run at weekends. This mode of operation is useful when large numbers of variations of an instrument design are to be evaluated (see the example discussed in Section IV). In many instances, 10-12 hours of CPU time is enough to obtain an answer with sufficient statistical accuracy and longer runs can be obtained overnight by splitting each variation across multiple machines.

In other cases however, it is desirable to run quasi-interactively. For example, when exploring the feasibility of a new instrument design or undertaking an optimization that requires considerable human intervention and decision making, it is useful to be able to obtain rapid feedback. To support this way of working, we allow jobs to run during normal office hours provided they meet certain criteria.

Condor is not particularly designed for rapid job turn-around. For example, whilst a 5-minute response time for a 500 min MC simulation CPU job should be possible simply by splitting the job across 100 machines, a naïve implementation of this splitting proves inefficient. **Figure 3** plots the number of started and completed jobs as a function of time for 2 configurations:

1. Naïve implementation; job submission is performed directly from user's PC.
2. Server implementation; executable and input files are uploaded to a central server, from where they are copied separately by each host machine.

It can be seen that there is a significant execution bottleneck for implementation 1, with the last jobs not commencing until at least 6 minutes from submission. This

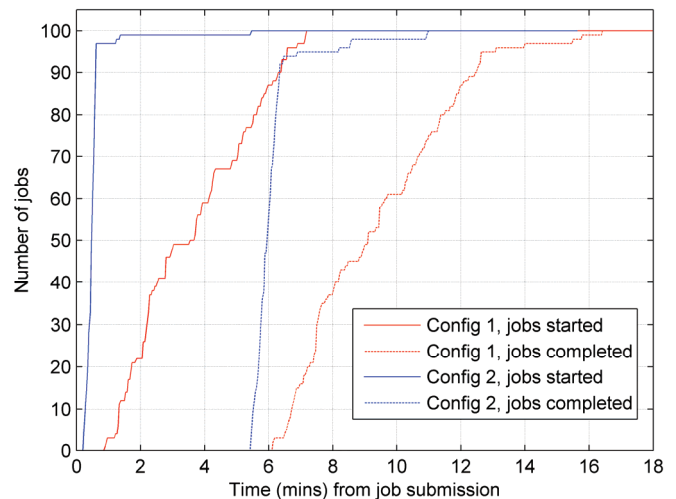


Fig. 3 Time profiles for numbers of started and completed jobs for 2 different configurations. In each case, 100 five minute jobs were submitted.

delay results from the time taken to transfer the problem input and executable files (a total of 1.9 MB in this example) from the user's PC to each host machine. Copying these files in advance to a server significantly reduces this block as only a trivially small script file needs to be transferred to each host machine. In configuration 2, the majority of jobs are started within one minute of submission and more than 90% are completed within 6.5 minutes.

In both configurations, there is a tail of jobs that take a few extra minutes to complete. Inspection shows that these are jobs evicted from host machines when a user reclaims their desktop PC. This tail can be minimized by instructing Condor to preferentially chose hosts machines that have been idle for an extended period of time.

Careful attention to file transfers is necessary to obtain satisfactory performance. However, with appropriate optimization, large numbers of jobs can be run with a non-CPU time overhead of 1-2 minutes.

### IV. Example - X-ray Diffraction Analyser

Knowledge of process stream mineralogy is important for monitoring and controlling minerals-processing plants. We have developed an on-stream instrument that uses energy-dispersive X-ray diffraction (EDXRD) to carry out quantitative mineralogical analysis on slurries.<sup>13)</sup> The analyser must determine the proportions of key mineral phases, present as finely ground solid particles suspended in water.

Bragg's law states that coherent scattering occurs for X-rays whose wavelength  $\lambda$  satisfies  $n\lambda=2d\sin\theta$  where  $n$  is an integer,  $d$  is the crystal-plane spacing and  $\theta$  is the scattering angle. The EDXRD technique measures the energy spectrum of X-rays that have been coherently scattered through a fixed angle from crystal planes of mineral grains present in a material. From the Bragg equation, peaks in the energy spectrum correspond to scattering from particular crystal planes, with the peak energy giving the  $d$ -spacing value and the peak amplitude related to the amount of the corresponding mineral present.

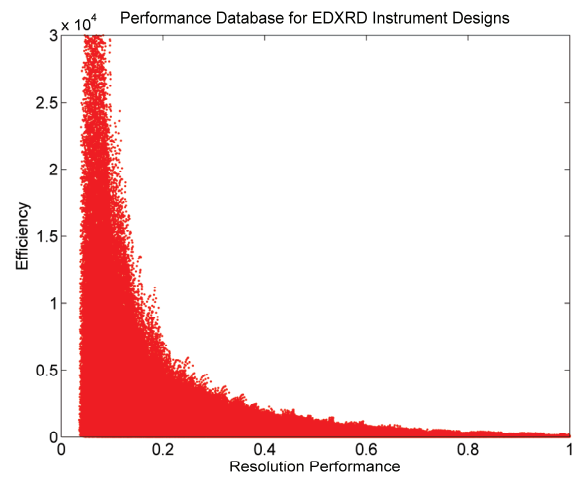
An EDXRD analyser consists of a polychromatic X-ray source, typically an X-ray tube, an energy-resolving X-ray detector and a set of collimators to define incident and scattered X-ray beams. There is a fundamental trade-off between instrument resolution – the ability to resolve crystal planes with similar  $d$ -spacings – and efficiency, namely the ability to measure the diffraction spectrum in a short time. Achieving high resolution requires narrow collimator openings to define a narrow range of scattering angles, whereas efficiency requires wide openings to accept more incident and scattered X-rays.

The resolution-efficiency tradeoff is not controlled by a single parameter, but depends in a complex way on the incident and scattered beam collimators, the X-ray scattering angle, sample presentation and source operating parameters. The optimal solution also varies from application to application, depending on the mineralogical makeup and X-ray attenuation properties of the slurry to be measured.

To address this problem, we developed a database of performance parameters (resolution and efficiency) for a large number of EDXRD design variations. These performance numbers were estimated using MC simulation of each design using a modified version of the EGSnrc code.<sup>14)</sup> In total, 28,125 designs were simulated. The XPERT interface was used to set up the design permutations using the ‘variations’ capability to map a unique permutation number to the desired geometry design. The jobs were submitted to Condor for processing using the XPERT interface. Each simulation was run for 300 minutes, which represents a total computation time of approximately 16 years. The jobs were run overnight in batches of 125; approximately 1,000 simulations were run each week. The entire database was completed in approximately 6 months. **Figure 4** shows the results obtained. Each point represents the performance of an individual instrument design permutation. Efficiency and resolution (arbitrary units) are plotted, with large values indicating better performance. The results show that there is a family of Pareto-optimal designs lying along the upper-right edge of the distribution that provide the best trade-off between resolution and efficiency. The best instrument design for any application is chosen from this family.

To help select the optimal design for a given application, a MATLAB<sup>®15)</sup> code and graphical interface were developed. These allow the user to specify measurement and instrument parameters, such as the material to be analyzed, X-ray tube settings, acceptable resolution limits and instrument geometry restrictions. The code uses these parameters to select the designs from the database that best satisfy the specified requirements. The diffraction spectrum of the sample material produced by each of these designs is modeled and can be displayed graphically. Designs selected by the user can be imported directly back into XPERT for more detailed investigation.

The XPERT interface and Condor computing environment were essential for creating and running the large number of design variations needed to establish this database.



**Fig. 4** Performance data for the designs in the EDXRD instrument optimizer. A good design has both high resolution performance and efficiency.

## V. Conclusions

Monte Carlo simulation techniques are a powerful tool for designing and optimizing nuclear instruments. Mature codes are available in the public domains that are capable of accurately simulating the underlying physics of radiation transport and interactions. We have described tools to:

- Facilitate the rapid setting up of complex instrument models;
- Creating and managing large numbers of design permutations;
- Rapidly running simulations of these permutations on a large cluster of desktop PCs; and
- Identify the design permutations with optimal performance.

These tools are in regular use by our group to design measurement and imaging instruments using X-rays, gamma-rays and neutrons to solve a wide range of problems in the minerals and security industries.

## References

- 1) X-5 Monte Carlo Team, *MCNP – A General Monte Carlo N-Particle Transport Code, Version 5*, LA-UR-03-1987, Los Alamos National Laboratory (LANL) (2003).
- 2) D. B. Pelowitz, *MCNPX User’s Manual, Version 2.6.0*, LA-CP-07-1473, Los Alamos National Laboratory (LANL) (2008).
- 3) I. Kawrakow, D. W. O. Rogers, *The EGSnrc Code System: Monte Carlo Simulation of Electron and Photon Transport*, NRCC Report, PIRS-701 (2003).
- 4) J. Allison *et al.*, “Geant4 developments and applications,” *IEEE Trans. Nucl. Sci.*, **53**[1], 270-278 (2006).
- 5) Condor High Throughput Computing website, <http://www.cs.wisc.edu/condor>
- 6) J. R. Tickner, G. J. Roach, “PHOTON – an Optical Monte Carlo code for Simulating Scintillation Detector Responses,” *Nucl. Instr. Meth. Phys. Res.*, **B263**[1], 149-155 (2007).
- 7) J. R. Tickner, “XPERT – a system for geometry definition, verification and visualization,” *Proc. Monte Carlo 2005*, Apr. 17-21, 2005, Chattanooga, USA (2005).



- 8) J. K. Tuli, *Evaluated Nuclear Structure Data File – A manual for preparation of data sets*, BNL-NCS-51655-01/02-Rev (2001).
  - 9) R. Bencardino, G. Roach, J. Tickner, “Efficient Monte Carlo simulation of delayed activation analysis experiments,” *Nucl. Instr. Meth. Phys. Res.*, **B268**, 513-518 (2010).
  - 10) D. J. S. Findlay, “Analytic representation of Bremsstrahlung spectra from thick radiators as a function of photon energy and angle,” *Nucl. Instr. Meth. Phys. Res.*, **A276**, 598-601 (1989).
  - 11) D. Tucker, G. Barnes, D. Chakraborty, “Semiempirical model for generating tungsten target x-ray spectra,” *Med. Phys.*, **18**[2], 211-218 (1991).
  - 12) G. J. Roach, J. R. Tickner, “Automated Instrument Design and Optimisation using a Genetic Algorithm Running on a Distributed Processing Network,” *Nucl. Instr. Meth. Phys. Res.*, **B263**[1], 156-162 (2007).
  - 13) J. O’Dwyer, J. R. Tickner, “Quantitative mineral phase analysis of dry powders using energy dispersive X-ray diffraction,” *Appl. Radiat. Isot.*, **66**[10], 1359-1362 (2008).
  - 14) J. N. O’Dwyer, J. R. Tickner, “Modelling diffractive X-ray scattering using the EGS Monte Carlo code,” *Nucl. Instr. Meth. Phys. Res.*, **A580**[1], 127-129 (2007).
  - 15) MATLAB®, <http://www.mathworks.com>
-