

# iMASS: Computational NRF Spectra Signal from Geant4

John Perry, Shanjie Xiao, Tatjana Jevremovic  
School of Nuclear Engineering, Purdue University, West Lafayette, IN 47907  
{joperry, xiaosj, tatjanaj} @purdue.edu

## Abstract

*A primary objective of the intelligent Model-Assisted Sensing System (iMASS) project is to combine computer simulated nuclear resonance fluorescence (NRF) with real-time Monte Carlo calculations to be used in the detection of nuclear materials hidden in cargo containers. This paper introduces the preliminary results related to the Monte Carlo simulation of NRF spectra signal and the importance of reconfigurable computational technique to accelerate Monte Carlo iMASS computational sequences. The selected Monte Carlo platform Geant4 was analyzed for the NRF implementation, modeling of a 3.2MV bremsstrahlung spectrum and the radiography images of cargo container.*

## 1. Introduction

Nuclear resonance fluorescence (NRF) is the process of resonant nuclear absorption of a photon, followed by de-excitation and emission of one or more photons. The resonance fluorescence cross section, without considering Doppler broadening, is described with a Breit-Wigner formula:

$$\sigma(E) = \pi \left( \frac{\lambda}{2\pi} \right)^2 \frac{2J_1 + 1}{2J_0 + 1} \frac{\Gamma_0 (\Gamma/2)}{(E - E_r)^2 + (\Gamma/2)^2} \quad (1)$$

where  $\lambda$  and  $E$  are the wavelength and energy of the incident photon,  $E_r$  is the resonance energy of a nucleus,  $J_0$  and  $J_1$  are nuclear spins of the ground and excited states,  $\Gamma$  is the total decay width of the excited state, and  $\Gamma_0$  is the partial width of the electromagnetic de-excitation. NRF is an excellent method to probe nuclear materials, since each isotope has its own specific NRF spectrum signature. This signature creates a unique thumbprint that can be used to identify otherwise ambiguous cargo content.

## 2. Monte Carlo simulation of NRF

A primary objective of the iMASS is to combine computer simulated NRF with real-time Monte Carlo calculations. Although the physics of NRF has been studied for many years, the Monte Carlo (MC) simulation

of NRF became recently of interest to develop and implement into the existing Monte Carlo codes [1].

### 2.1. Selection of Monte Carlo platform

There are many MC codes platforms each with their own strengths and weaknesses. Most used at least in US are the Monte Carlo N-Particle Transport 5 (MCNP5) [2, 3] and Geometry And Tracking 4 (Geant4) [4, 5], and therefore considered as top candidates for the iMASS system development. Geant4 was chosen over MCNP5 due to its open source (thus the program is free, no license) modularity. Additionally, Geant4 has great visualization tools and user interface attributes.

Geant4 is programmed in C++ allowing users to take advantage of object-oriented programming abilities. For example, a user can compile a benchmark and individually change small parameters of that benchmark, even as the benchmark runs, without recompiling. The program offers great flexibility and control for modeling an environment, physics, particle tracking, and particle visualization. The only downside of this flexibility is that the learning curve for the program is much more intense than an input deck driven modeling code. The user must actually encode (in C++) the modeling environment, particle setup, and tracking. Figure 1 shows a generic flowchart for Geant4 programming.

A major strength of Geant4 is its modularity and selection regarding physics processes. The user has the ability of including and modifying a full range of nuclear and magnetic physics. Although as reported in [1] the NRF has been introduced into the Geant4 (with pure accuracy of the produced NRF signal) we will develop our own module.

The visualization capabilities of Geant4 are also very desirable and easy to use. The graphics can be generated without hardware acceleration in an OpenGL environment. The visualization capability also supports other graphical processes apart from OpenGL. The graphical output can be controlled by a string of commands or even through a user interface (if designed by the user). The particles can be viewed in a "real-time" sense as the simulation runs. Figure 2 demonstrates some of the graphical capabilities of Geant4. This figure is a

model of the ATLAS detector system used at the Large Hadron Collider facility. With Geant4, it is possible to make robust images with real time particle modeling easily and effectively.

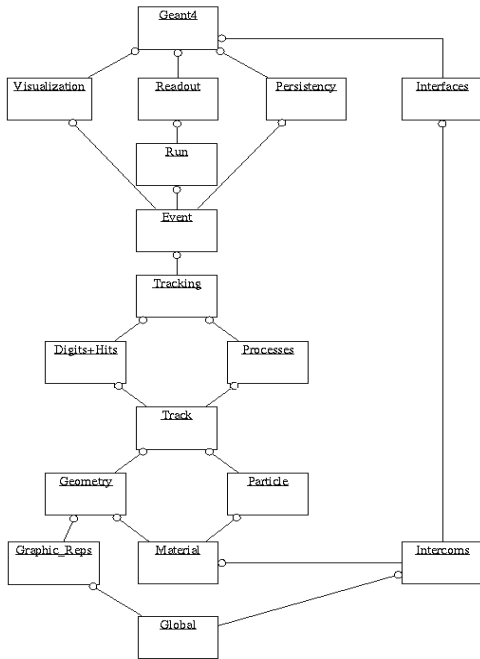


Figure 1. Geant4 modeling flowchart [5]

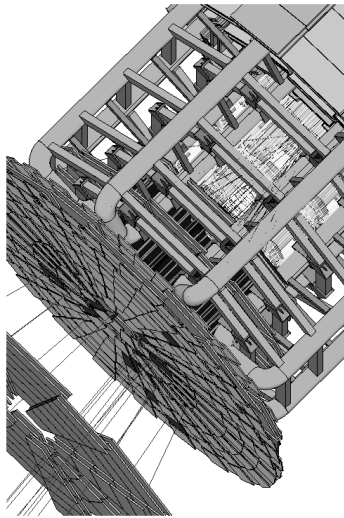


Figure 2. Geant4 visualization example [5]

## 2.2. Geant4 initial benchmark

Our first benchmark prior to the NRF module development is developed to create the main Geant4 framework for the iMASS system development. The NRF real-time cargo imaging will be based on the NRF source

that is usually the bremsstrahlung gamma spectrum. We have selected the NRF source as specified in [6].

The Geant4 benchmark consists of a 3.2 MeV electron beam bremsstrahlung tungsten source. The tungsten is 2 millimeters thick. The beam is oriented so that the electrons struck the tungsten foil perpendicularly, and the X-rays are obtained from the opposite side of the foil.

The tungsten is surrounded by a lead shield with a 1 cm x 1 cm window. The main objective was to obtain the X-ray spectrum at the desired point of interest approximately 40 cm from the X-ray's point of origin. This 40 cm location is where the sample target, such as highly enriched uranium, would be located. Other parameters of the model include: sample target dimension of 1.9 mm x 3 cm x 3 cm, the sample target is encased in polyethylene terephthalate, and the sample target material could be changed to another type, such as manganese. Figures 3 and 4 represent the setup of the actual experiment and its Geant4 model.

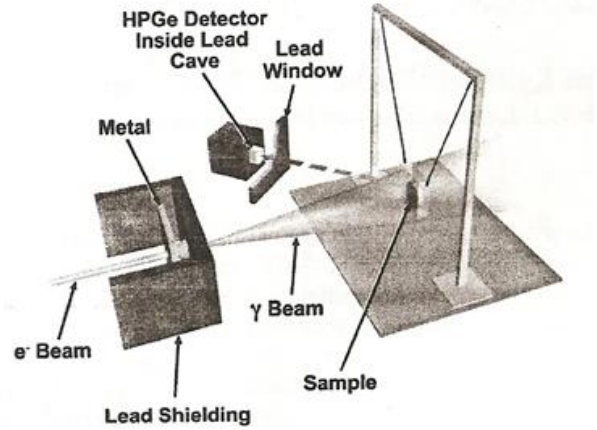


Figure 3. Actual experimental setup from [6]

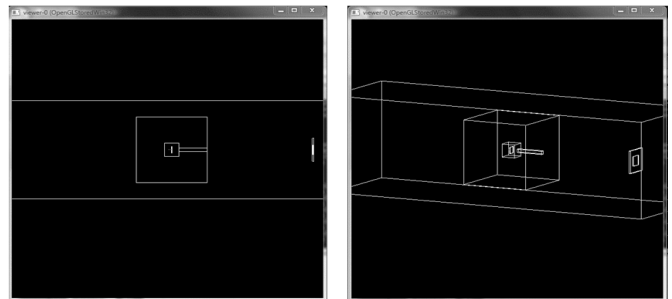
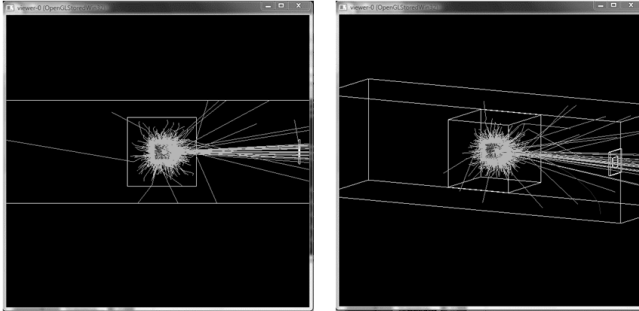


Figure 4. Geant4 model of experiment shown in Figure 3

The visualization capabilities of Geant4 are very useful in verifying the setup of the model. Ten thousand particles were tracked in a sample run to qualitatively examine the direction of the generated X-rays. Figure 5 displays the output of the sample run in the X-Z plane as well as a skewed 20 degree clockwise rotation of the X-Z plane.

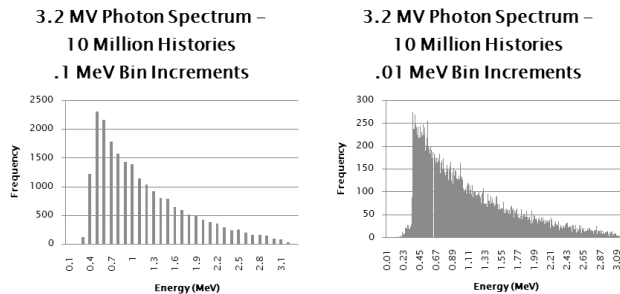
In the following figure, the particle is initialized as an electron at 3.2 MeV. That electron is located in the center of the central box. The electron vector is perpendicular and incident to the tungsten bremsstrahlung target. The daughter bremsstrahlung X-rays are tracked based upon their energy and path vector. The X-rays are filtered through a lead collimator with a window. The collimated X-ray is directed to a final target as shown in Figure 3.



**Figure 5. Geant4 X-ray model**

The X-ray energy spectrum at the target (labeled as the sample in Figure 3) is computed. Approximately two photons are registered at the final target location for every thousand histories generated. Each history included the initial event electron, its scattering, and the subsequent paths of the daughter X-rays. Python, a computational language, was used to create a tool to parse the output information from Geant4 and obtain the resulting X-ray spectrum.

Figure 6 shows the resulting bremsstrahlung spectrum which consists of twenty thousand accumulated data points. Ten million electrons were initialized with only a small fraction of the collimated X-rays reaching the target to be analyzed. The shape of the bremsstrahlung continuum is similar to another bremsstrahlung spectrum found in [7]. The photon spectrum can be smoothed by recomputing with a greater amount of histories, or a numerical method such as interpolation and spectra erosion can be applied.



**Figure 6. 3.2MV Geant4 bremsstrahlung spectrum for geometry shown in Fig.4**

### 3. Hardware acceleration of Geant4

To realize the real-time iMASS system, we will develop custom-made hardware acceleration. A specific hardware chip will be designed based on the reconfigurable computation technique. While not implementing computation on general purpose chips, reconfigurable computation has specifically designed hardware for each specific algorithm so as to get better performance; it does not have the high cost and high risk of application-specific integrated circuit (ASIC). It came with the advancement of the silicon device field programmable gate array (FPGA).

FPGA is an advanced digital chip consisting of numbers of programmable logic blocks based on look up table (LUT) and switch matrices. The logic function of each logic block and the interconnection of logic blocks can be configured by users for their expected circuits. The cost of FPGA chip is much lower than the tapeout cost of ASIC and the development time is also shorter. Although the clock frequency available on FPGA chip is lower than clock of general CPU (e.g. 550MHz on latest Xilinx Virtex-5 while over 3GHz on Intel CPU), multiple parallel algorithm blocks can be built on one FPGA chip. Compared with the multi-core parallelism in general CPU, reconfigurable computation on FPGA chip supplies parallelism on a smaller granularity and the specifically designed architecture can increase its efficiency.

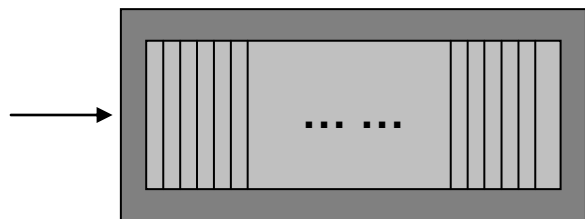
The reconfigurable computation technique has been applied to many fields: seismic [8], encryption [9], N-body simulation [10], molecular dynamics [11], DNA analysis [12], etc.. The acceleration times were from 10 to over 1000. Some researchers have tried to accelerate the application of Monte Carlo simulation in other fields. As reported in [13] the Monte Carlo radioactive heat transfer simulation was accelerated about 10 times by speeding up the geometry module using the FPGA. As described in [14] the FPGA is applied to speed up the Monte Carlo simulation for the BGM interest rate estimates for pricing derivatives. They built multiple hardware components, e.g. fast division, Gaussian noise generator and path generation, and achieved 25x speedup.

### 4. Time profile analysis of Geant4 simulations

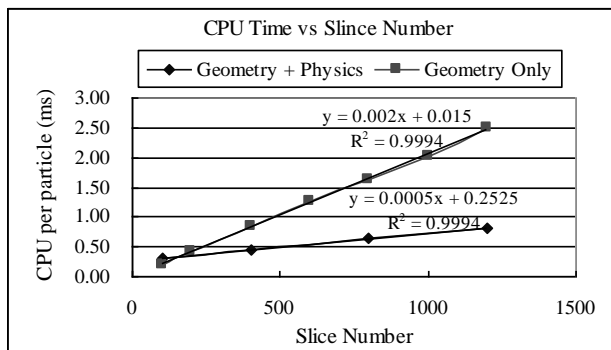
In order to accelerate an algorithm, its time profile must be analyzed at first in order to identify the most time consuming sections. For the Geant4 time profile we used Callgrind, which is a component of Valgrind. This tool can nicely identify instructions executed for each line of the source code with per-function, per-module and whole-program summaries, plus to give much of extra information about call-graphs. In addition, we used the visualization tool Kcachegrind to graphically show the time profile results.

### 4.1. Basic application of Geant4

In the preliminary stage, we first analyzed the time profile of basic Geant4 applications. A simple study model is based on an 8MeV mono-energy gamma beam vertically impinging from the left side of a standard 40' cargo container (as shown in Figure 7). In the basic test, the cargo is filled with water and is subdivided into a number of slices. We profiled the Geant4 for two types of tests: an analysis of geometry only without including the physics interactions and an analysis of geometry including the electromagnetic interactions. Figure 8 shows the CPU time for these two types of tests.



**Figure 7. Cargo container configuration for the Geant4 basic application CPU time profile test**

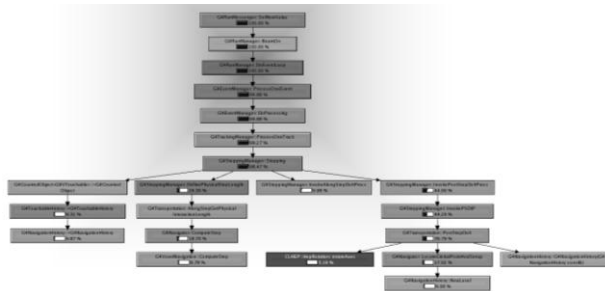


**Figure 8. CPU time at different number of slices inside the cargo container**

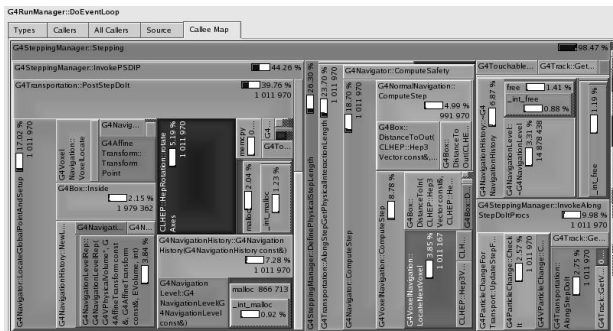
In both tests, the relation between CPU time and the number of slices is linear. The reason for this is because when the slice number increases, the thickness of each slice decreases and thus the number of slices passed by a particle will increase almost linearly regardless the physics interactions are included or not. On the other hand, with physics interactions, the input particle (and its secondary radiations) will be stopped within an amount of slices; the particle will pass through all slices if physics interactions are not considered. Thus, when the slice number is small, each simulation step considering physics interactions will cost more CPU time, while when the slice number is large, a particle in the geometry only simulation will pass much more slices compared to the test that includes the physics simulation and the CPU time is longer. This conclusion is based on the simple model

(Figure 7) that includes the subdivision of the interior of the cargo along one dimension only. If a 3D grid was applied, the required memory to define all voxels will be in the order of  $O(n^3)$ , but the size of each dimension of a voxel will still be proportional to  $1/n$ , giving the running time of the order of  $O(n)$ .

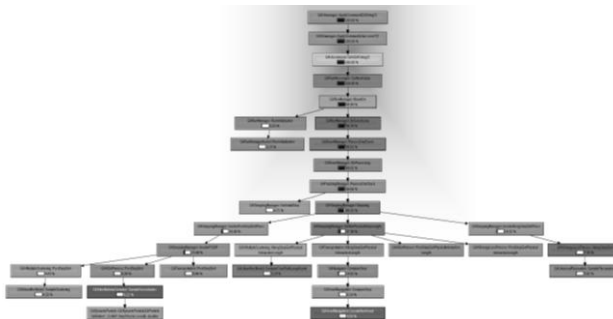
Figure 9 and Figure 10 show the call-graph and the time profile of a simulation without considering physics interactions, while Figure 11 and Figure 12 show the call-graph and time profile of a simulation with physics interactions. There are more functions called in Figure 11 and Figure 12 due to the consideration of various physics interactions.



**Figure 9 Call-graph of a simulation without considering physics interactions**



**Figure 10. Time profile of a simulation without considering physics interactions**



**Figure 11. Call-graph of a simulation with electromagnetic interactions**

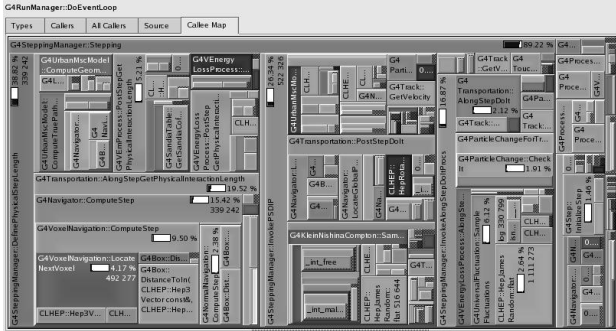


Figure 12. Time profile of a simulation with electromagnetic interactions

## 4.2. Radiography simulation in Geant4

This section describes the Geant4 time profile for a more complicated simulation. A radiography of a 40' cargo container with four tubes made of different materials, Al, Mn, 97% enriched uranium and 5% enriched uranium, and one water cylinder located at the center, is modeled (Figure 13).

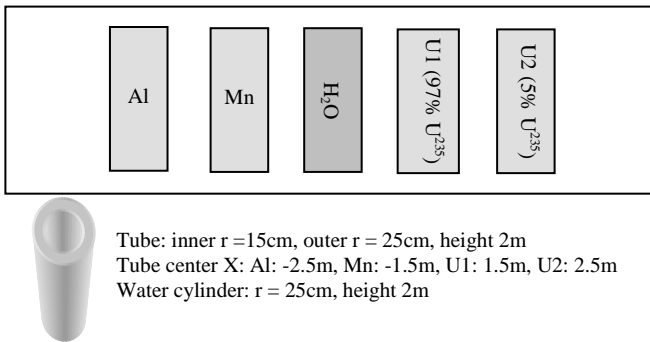


Figure 13. Cargo container configuration for Geant4 radiography simulation



Figure 14. Radiography image generated by Geant4

The Geant4 radiography image is shown in Figure 14. The image consists of 500 x 100 pixels, i.e. each pixel is about 2.4cm x 2.4cm. 200 million primary particles which were sampled according to the 3.2MV bremsstrahlung spectrum were simulated. In the generated figure, the lighter region had higher particle count while darker region had lower count. Figure 15 and Figure 16 show the time profile and call-graph of the radiography simulation. It can be seen that 75% time was spent on computing the moving steps (the “Stepping” function), while in the basic

test described in Section 4.1, this ratio is around 90%. This difference is because more time was spent on initialization and sampling the radiography result.

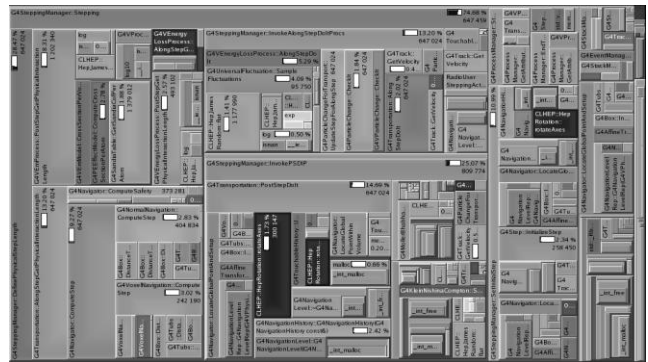


Figure 15. Time profile of the cargo container radiography simulation

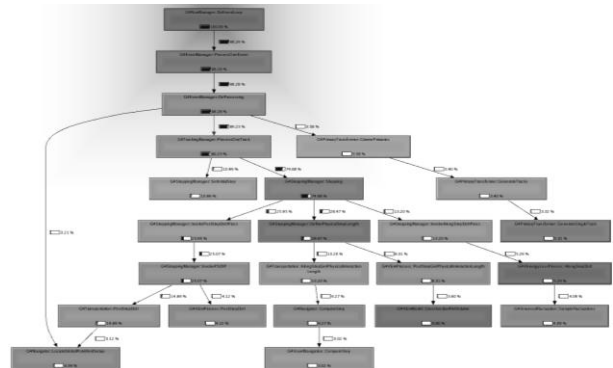


Figure 16. Call-graph of the radiography simulation

## 5. Discussion and future work

The Monte Carlo simulation of NRF is the essential part of the iMASS system. We selected Geant4 as the development platform because it is an open-source, modularized, powerful Monte Carlo code. Two applications related to the iMASS system, bremsstrahlung spectrum computation and radiography simulation, have been developed based on Geant4. The computation of the bremsstrahlung spectrum will be used in the development and testing of the NRF module, and the information from radiography image will be used to develop the pattern recognition technique based on the gray scale imaging. The time profile of Geant4 computations was analyzed. The most time-consuming part is computing the moving steps, 75%~90% of total CPU time in the analyzed examples.

The development of the NRF module is our next task. Important objectives for building this module include: algorithms for the control of Geant4, a database of lines specific to NRF along with their specific cross-sections. Currently the nuclear database NuDat2, provided by BNL

[15], hosts a large amount of information including line energies, spin states, and line widths that are necessary for the calculation of an NRF cross-section. Additionally, as described in [1] an NRF module for Geant4 has been developed; it can be useful for the benchmark. To accelerate NRF computation, the algorithm must be researched to build HDL (hardware description language) modules. According to processed time profile analysis, the Geant4 contains quite a large number of small functions, which must be organized into these HDL modules for high performance. Thus an effective architecture, including instruction set, data flow, I/O format, etc., will be designed.

## References

- [1] D.V. Jordan and G.A. Warren, "Simulation of nuclear resonance fluorescence in Geant4," *Proc. NSS '07*, 2007, pp. 1185-1190.
- [2] MCNP Team, "MCNP Home Page," 2005; <http://mcnp-green.lanl.gov/>.
- [3] MCNP Team, *MCNP 5.1.40 RSICC Release Notes*, LANL, 2005.
- [4] S. Agostinelli, et al., "Geant4 - a simulation toolkit," *Nuclear Instruments and Methods in Physics Research A*, vol. 506, 2003, pp. 250-303.
- [5] CERN, "Geant4: A toolkit for the simulation of the passage of particles through matter," 2008; <http://geant4.web.cern.ch/geant4/>.
- [6] G.A. Warren, et al., "Nuclear Resonance Fluorescence of  $^{235}\text{U}$ ," *Nuclear Science Symposium Conference Record*, 2006. *IEEE*, vol. 2, 2006, pp. 914-917.
- [7] V. Maniquis, "Monte Carlo Dose Verification of an X-Ray Beam in a Virtual Water Phantom," Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, 2006.
- [8] C. He, et al., "Accelerating Seismic Migration Using FPGA-Based Coprocessor Platform," *Proc. 12th Annual IEEE Symposium on FCCM'04*, 2004.
- [9] T. Ristimaki and J. Nurmi, "Implementation of a fast 1024-bit RSA encryption on platform FPGA," *Proc. 6th IEEE International Workshop on DDECS'03*, 2003.
- [10] G. Lienhart, et al., "Using Floating-Point Arithmetic on FPGAs to Accelerate Scientific N-Body Simulations," *Proc. 10th Annual IEEE Symposium on FCCM'02*, 2002.
- [11] J. Villarreal, et al., "Compiled Code Acceleration of NAMD on FPGAs," *Proc. RSSI'07*, 2007.
- [12] J.D. Buhler, et al., "Mercury Blastn: Faster DNA Sequence Comparison Using a Streaming Hardware Architecture," *Proc. RSSI'07*, 2007.
- [13] M.B. Gokhale, et al., "Accelerating Monte Carlo Radiative Heat Transfer Simulation on a Reconfigurable Computer: An Evaluation," *Proc. FPL 2004*, 2004.
- [14] G.L. Zhang, et al., "Reconfigurable Acceleration for Monte Carlo based Financial Simulation," *Proc. IEEE 2005 Conference on Field-Programmable Technology*, 2005.
- [15] BNL, "NuDat 2.4," 2008; <http://www.nndc.bnl.gov/nudat2>.