Epp: A C++ EGSnrc user code for x-ray imaging and scattering simulations

Jonas Lippuner
Department of Medical Physics, CancerCare Manitoba, 675 McDermot Avenue, Winnipeg, Manitoba R3E 0V9, Canada and Department of Physics and Astronomy, University of Manitoba, Winnipeg, Manitoba R3T 2N2, Canada

Idris A. Elbakri[a)]
Department of Medical Physics, CancerCare Manitoba, 675 McDermot Avenue, Winnipeg, Manitoba R3E 0V9, Canada; Department of Physics and Astronomy, University of Manitoba, Winnipeg, Manitoba R3T 2N2, Canada; and Department of Radiology, University of Manitoba, 820 Sherbrook Street, Winnipeg, Manitoba R3A 1R9, Canada

Congwu Cui and Harry R. Ingleby
Department of Medical Physics, CancerCare Manitoba, 675 McDermot Avenue, Winnipeg, Manitoba R3E 0V9, Canada

(Received 26 February 2010; revised 25 January 2011; accepted for publication 25 January 2011; published 1 March 2011)

Purpose: Easy particle propagation (Epp) is a user code for the EGSnrc code package based on the C++ class library egspp. A main feature of egspp (and Epp) is the ability to use analytical objects to construct simulation geometries. The authors developed Epp to facilitate the simulation of x-ray imaging geometries, especially in the case of scatter studies. While direct use of egspp requires knowledge of C++, Epp requires no programming experience.
Methods: Epp's features include calculation of dose deposited in a voxelized phantom and photon propagation to a user-defined imaging plane. Projection images of primary, single Rayleigh scattered, single Compton scattered, and multiple scattered photons may be generated. Epp input files can be nested, allowing for the construction of complex simulation geometries from more basic components. To demonstrate the imaging features of Epp, the authors simulate 38 keV x rays from a point source propagating through a water cylinder 12 cm in diameter, using both analytical and voxelized representations of the cylinder. The simulation generates projection images of primary and scattered photons at a user-defined imaging plane. The authors also simulate dose scoring in the voxelized version of the phantom in both Epp and DOSXYZnrc and examine the accuracy of Epp using the Kawrakow–Fippel test.
Results: The results of the imaging simulations with Epp using voxelized and analytical descriptions of the water cylinder agree within 1%. The results of the Kawrakow–Fippel test suggest good agreement between Epp and DOSXYZnrc.
Conclusions: Epp provides the user with useful features, including the ability to build complex geometries from simpler ones and the ability to generate images of scattered and primary photons. There is no inherent computational time saving arising from Epp, except for those arising from egspp's ability to use analytical representations of simulation geometries. Epp agrees with DOSXYZnrc in dose calculation, since they are both based on the well-validated standard EGSnrc radiation transport physics model.
[DOI: 10.1118/1.3555296]

Key words: Monte Carlo simulation, EGSnrc, DOSXYZnrc, scatter, imaging simulation

I. INTRODUCTION

Monte Carlo simulations represent a useful and accurate method for modeling particle transport in medical imaging and radiation therapy. For example, Monte Carlo simulations have been used recently to model and devise correction schemes for scattered photons in cone-beam computed tomography.[1,2] The EGSnrc Monte Carlo package features validated models of particle (photon, electron, and positron) interactions and is a widely accepted standard for photon-electron transport.[3]

The EGSnrc package only implements particle transport through matter. User codes are required to implement comprehensive simulations including particle sources and propagation geometries. DOSXYZnrc, for example, is an extensively used EGSnrc user code for three-dimensional dose calculations.[4] The EGSnrc C++ class library (egspp) (Ref. 5) provides tools to model complex geometries and sources, including analytically defined phantoms that could result in reduced computational time. Direct use of the egspp library requires C++ programming experience and the development of user codes.

In imaging simulations, the user is usually interested in an output image. For example, in simulating a cone-beam x-ray system, photons exiting the object need to be propagated to a detector, where an image is formed. We developed easy particle propagation (Epp), a user code based on the egspp package, to facilitate the performance of x-ray imaging simulations.[8] Epp provides a user-friendly interface to egspp and some useful additional features. Epp can be used out-of-the-box and requires no additional programming. In this brief technical note, we discuss the features of Epp and present simulation results that illustrate its functionality.

II. MATERIALS AND METHODS

II.A. Epp features

In Epp, the parameters of the Monte Carlo simulation and geometry are defined in an input file using the egspp input file format. The input file includes the simulation geometry, particle source, image plane, and various other simulation parameters. Epp adds two additional features to the egspp input file format. It is possible to directly use a voxelized DOSXYZnrc geometry by referring to a phantom file (*.egsphant file). Epp also introduces a mechanism for referring to existing geometry files and other input or simulation control files within any Epp input file. Therefore, complex geometries, particle sources, and simulation parameters can be defined in separate files and can be referred to by the main control input file. Epp is run from the command line.

Through either a command line or input file commands, the user can specify whether he/she wishes for the simulation to generate images of primary photons, single Compton scatter, single Rayleigh scatter, multiple scatter, and/or all photons reaching a predefined imaging plane. The imaging plane is a user-defined pixelated "virtual detector," where a photon-count and/or energy fluence per pixel is determined. This virtual detector is simply a plane in the geometry where particle tracking stops and where the number of photons or energy fluence is recorded. It is of course possible to model particle transport through detector materials such as scintillators, but we do not report on such results in this note. Image data, whether photon counts or energy fluence, are stored in binary files that can be easily read for data analysis. For a quick inspection of the results, Epp can generate bitmap images. Epp may be run in either single process mode or parallel batch mode, similar to other EGSnrc user codes. In parallel batch mode, all results will be combined into single output files automatically after all processes finish. Epp does not store the phase space photon information unless specified by the user.

The egspp library can score the deposited dose. For a detailed distribution of deposited dose, analytical representations of the geometry are cumbersome. Epp simplifies this step with its ability to incorporate a voxelized phantom in the simulation geometry. Unlike DOSXYZnrc, Epp does not normalize the dose with respect to the incident particle fluence.

TABLE I. EGSnrc simulation parameters for DOSXYZnrc and Epp.

Global ECUT	0
Global PCUT	0
Global SMAX	1e10
ESTEPE	0.25
XIMAX	0.5
Boundary crossing algorithm	EXACT
Skin depth for BCA	0
Electron-step algorithm	PRESTA-II
Spin effects	On
Brems angular sampling	Simple
Brems cross sections	BH
Bound Compton scattering	Off
Pair angular sampling	Simple
Photoelectron angular sampling	Off
Rayleigh scattering	On
Atomic relaxations	On
Electron impact ionization	On

II.B. Simulations

To illustrate the "imaging" features of Epp, we simulated a monoenergetic 38 keV point source irradiating a cylindrical water phantom, 12 cm in diameter, embedded in a $12.8 \times 12.8 \times 12.8$ cm^3 air cube. We used 10^{10} photon histories. The source and the image plane were at 25 cm from the center of the phantom. The virtual detector consisted of 512×5121 1-mm pixels. The incident beam was collimated to the face of the air cube. Other EGSnrc simulation parameters are listed in Table I. We generated photon-count images of primary, single Compton, single Rayleigh, and multiple scatter photons, using both analytical and voxelized (64^3 2 mm voxels) versions of the phantom.

For comparing Epp dose scoring with DOSXYZnrc, we simulated 10^8 photon histories for both user codes for the geometry described above. All simulation parameters were kept the same and were identical between Epp and DOSXYZnrc. We performed the Kawrakow–Fippel test on the resultant three-dimensional dose distributions to compare the accuracy of Epp relative to DOSXYZnrc.[7]

All simulations were carried out on a Linux computer with eight Intel® Xeon® CPUs (X5460) with a clock frequency of 3.16 GHz and a total of 16 GB shared memory. The simulations reported herein were not parallelized and were performed on a single core.

III. RESULTS AND DISCUSSION

Figure 1 shows the projection photon-count images generated with Epp using the analytical geometry. The images from the voxelized geometry are similar and are not shown here. The average relative differences between the analytical and voxelized geometries are less than 1% in all cases. With dose scoring turned off, the analytical simulation is about 30% faster. Epp does not cause the simulations to run faster. The ability to represent a geometry analytically in egspp may

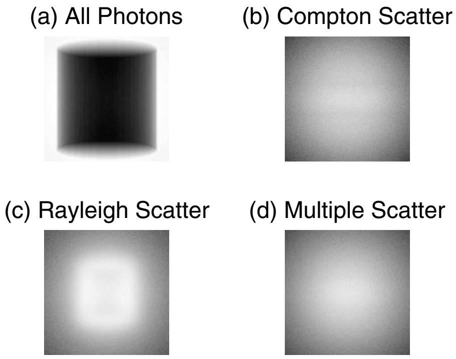

FIG. 1. Photon-count images of (a) all photons, (b) single Compton, (c) single Rayleigh, and (d) multiple scatter impinging on the imaging plane in the Epp simulation of an analytically represented geometry of a water cylinder. The image in panel (a) is slightly magnified relative to the other panels because the area in the shadow of the collimator that would appear totally dark has been cropped. The images are displayed on a log scale to enhance visibility.

result in computational savings, as it did in this case with a simple cylindrical object. This will vary with the complexity with which the geometry is represented.

Figure 2 shows the dose distributions in the central plane perpendicular to the axis of the water cylinder from Epp and DOSXYZnrc. The Epp results were normalized by the incident fluence (10^8 photos divided by the area of one side of the air cube). Figure 3 shows the dose profiles from Epp and DOSXYZnrc along the central row of the central slice, with error bars representing simulation uncertainties.

The histogram of differences in the three-dimensional dose distributions calculated by Epp and DOSXYZnrc is shown in Fig. 4. In the absence of systematic deviations, this histogram would be a realization of the normal distribution. Kawrakow and Fippel proposed a data fitting model to quantify systematic deviations, which in our case results in $\alpha_1 = 0.43$, $\alpha_2 = 0.027$, $\Delta_1 = -0.029$, and $\Delta_2 = 0.28$. In other words, 43% of voxels have a systematic deviation of 0.027 standard deviations and 2.7% have a systematic deviation of 0.28 standard deviations. Given that the combined uncertainty is about 0.09, this represents good agreement between the two user codes. Readers interested in more details of the Kawrakow–Fippel test are referred to the original paper.[7] The mean of the fit shown in Fig. 4 is −0.0049 and the standard deviation is 1.0025.

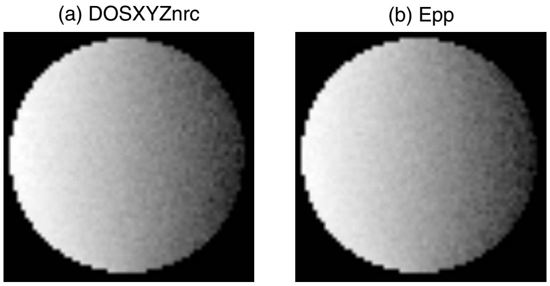

FIG. 2. Dose distributions in the central slice of the cylindrical phantom obtained from Epp and DOSXYZnrc.

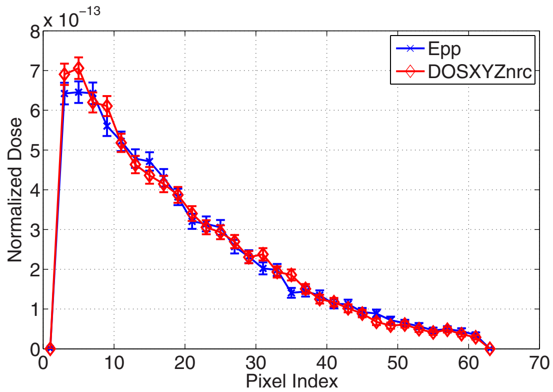

FIG. 3. Plots of the fluence-normalized dose profiles along the middle row of the central slice of the cylindrical phantom obtained from Epp and DOSXYZnrc. The error bars represent statistical uncertainty in the simulations.

The EGSnrc C++ class library enables the user to write user codes for EGSnrc using C++. Several user codes are discussed in the egspp manual.[5] Egspp also provides a geometry package that enables complex objects to be represented from simpler ones, such as boxes, spheres, cones, and planes and a set of particle sources. Epp is a user code for egspp that makes these features more accessible to the user without requiring C++ programming. As such, Epp does not directly impact the computational efficiency of the simulations and that is why we do not report extensively on computational time of the simulations we performed.

Epp uses the egspp scoring class to compute the dose deposited in a voxelized phantom. It is possible to compute the dose when a simulation is represented analytically. However, each analytical object would be considered a region and the simulation would return a single value for every region. In our case, we would have had a single deposited dose value for the water cylinder. By extending the egspp scoring class to voxelized geometries (where each voxel is now a region), Epp enables the user to obtain detailed dose distributions in

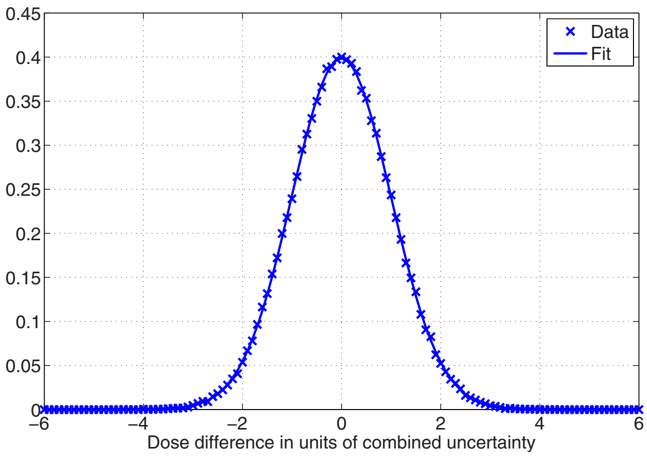

FIG. 4. Histogram of the dose differences (relative to the combined uncertainty) between Epp and DOSXYZnrc and a Gaussian fit as described in the Kawrakow–Fippel test. Fit parameters are $\alpha_1 = 0.43$, $\alpha_2 = 0.027$, $\Delta_1 = -0.029$, and $\Delta_2 = 0.28$.

an object of interest. Although our results showed good agreement in dose calculation between DOSXYZnrc and Epp, the reader is alerted to the fact that egspp appears not to be fully benchmarked and some inconsistencies have been observed.[6]

IV. CONCLUSIONS

Epp is a user code based on the EGSnrc C++ class library (egspp). The user code complements the features of egspp with the photon propagation to an imaging plane and the ability to use nested input files. The latter feature enables the user to construct complex geometries and simulations from simpler ones. In this technical note, we have illustrated the functionality of Epp.

Epp can be used for a variety of radiological imaging and radiation therapy applications. Epp is a free software (available at http://www.physics.umanitoba.ca/~elbakri/epp) and distributed under the terms of the General Public License, version 2 and any later version, as published by the Free Software Foundation.

ACKNOWLEDGMENTS

This work was supported in part by the CancerCare Manitoba Foundation.

---

[a] Electronic mail: idris.elbakri@cancercare.mb.ca

[1] R. Bhagtani and T. G. Schmidt, "Simulated scatter performance of an inverse-geometry dedicated breast CT system," Med. Phys. **36**, 788–796 (2009).

[2] Y. Chen, B. Liu, J. M. O'Connor, C. S. Didier, and S. J. Glick, "Characterization of scatter in cone-beam CT breast imaging: Comparison of experimental measurements and Monte Carlo simulation," Med. Phys. **36**, 857–869 (2009).

[3] E. Spezi and G. Lewis, "An overview of Monte Carlo treatment planning for radiotherapy," Radiat. Prot. Dosim. **131**, 123–129 (2008).

[4] B. Walters, I. Kawrakow, and D.W.O. Rogers, DOSXYZnrc Users Manual, National Research Council of Canada, Ottawa K1A 0R6, July 2009.

[5] I. Kawrakow, E. Mainegra-Hing, F. Tessier, and B. R. B. Walters, "The EGSnrc C++ class library," NRC Report No. PIRS-898 (rev A) (National Research Council of Canada, Ottawa, K1A OR6, July 2009).

[6] R. Behrens, "Inconsistencies in the egspp (the EGSnrc C++ class library) and in the SLAB module of BEAMnrc," Phys. Med. Biol. **55**, L33–L36 (2010).

[7] I. Kawrakow and M. Fippel, "Investigation of variance reduction techniques for Monte Carlo photon dose calculation using XVMC," Phys. Med. Biol. **45**, 2163–2183 (2000).

[8] C. Cui, J. Lippuner, H. R. Ingleby, D. N. M. Di Valentino, and I. Elbakri, "Epp: A C++ EGSnrc user code for Monte Carlo simulation of radiation transport," Proc. SPIE **7622**, 762251 (2010).