Epp - A C++ Monte Carlo simulation EGSnrc user code for dose calculation and imaging

Congwu Cui, Jonas Lippuner, Harry R. Ingleby, David N.M. Di Valentino, and Idris A. Elbakri

CancerCare Manitoba, Department of Medical Physics, 675 McDermot Ave, Winnipeg, MB R3E 0V9 Canada

ABSTRACT

Epp (Easy particle propagation) is a Monte Carlo simulation EGSnrc user code that we have developed for dose calculation in a voxelized volume, and to generate images for an arbitrary geometry irradiated by a particle source. The dose calculation aspect is a reimplementation of the function of DOSXYZnrc with the performance being significantly improved. The functionality can be readily extended to trace other kinds of particles.

Epp is based on the EGSnrc C++ class library (egspp) that offers great simplicity for setting up particle sources and simulation geometries, compared to DOSXYZnrc. Using an XML format for the input file makes the file parsing at least 1,000 times faster than the original input file format of egspp.

Compared to DOSXYZnrc, Epp is at least two times faster. Photon (other particles possible with minor extension to the current program) propagation to the image plane is integrated into Epp. When only the resultant images are needed, there is no need to save the particle data. This results in significant savings of data storage space, network load, and time for file I/O.

Epp was validated against DOSXYZnrc by comparing their simulation results with the exact same input. Epp can thus be used for faster Monte Carlo simulation for radiation dose applications and imaging applications.

Keywords: Monte Carlo simulation, x-ray scattering, radiation therapy, x-ray computed tomography, medical imaging

1. INTRODUCTION

Monte Carlo simulation is a useful tool in radiation imaging and therapy applications. For example, the need to perform Monte Carlo simulation to characterize photon interactions in x-ray imaging geometries, i.e., cone-beam computed tomography, has been rising steadily.[1–4] The need arises from the contribution of scattered photons that require correction to improve image contrast. Because the physics of particle (photon, electron, and positron) interaction with matter has been well built in the EGSnrc package, it is easy to build user applications based on it. DOSXYZnrc, based on the EGSnrc code system, is an extensively used Monte Carlo simulation tool in radiation therapy for three-dimensional dose calculations.[5] Although originally developed for dose calculations, it can be modified slightly to output information about all photons for radiation imaging and dose distribution studies.

Since the EGSnrc package does not model sources or geometries, it is functionally inconvenient when a particle source and/or geometry is complex. Although DOSXYZnrc implements a series of particle sources and a voxelized phantom geometry and works well with simple sources and objects, due to the input file format, the phantom (or object) can only have up to 9 different materials. Therefore, the EGSnrc C++ class library

Further author information: (Send correspondence to Idris A. Elbakri; E-mail: Idris.Elbakri@cancercare.mb.ca, Telephone: 1-204-787-2856)
Congwu Cui: E-mail: congwu.cui@cancercare.mb.ca, Telephone: 1-204-787-4133
Jonas Lippuner: E-mail: Jonas.Lippuner@cancercare.mb.ca
Harry Ingleby: E-mail: Harry.Ingleby@cancercare.mb.ca, Telephone: 1-204-787-2126
David DiValentino: E-mail: David.DeValentino@cancercare.mb.ca

(egspp) was developed to solve the problem of modelling more complex scenarios, in addition to providing other conveniences.[6]

For radiation imaging applications, photons need to be propagated to an image plane (i.e. an ideal X-ray flat panel detector). Although it can be done with a program written in a high-level computing language such as Matlab, when the photon and image counts are large, the computing time is extremely long. In addition, each file may be hundreds of gigabytes in size. As a result, it is difficult to save the files on conventional storage media. For data processing, the load for file I/O and data transfer across networks is also heavy. Because in most simulation applications, only images or dose distributions are needed, the intermediate simulation data can be processed on the fly right after it is generated so that only the needed results are saved. In some special cases if the intermediate simulation data is needed, it can be saved by specifying an option with a command line argument or in the input file.

To overcome the restrictions of DOSXYZnrc and also to improve the performance of the simulation, the user code Epp was developed based on the egspp package and the DOSXYZnrc MORTRAN user code. The new application package is written in C++, uses an XML input format, and performs faster in general. It can be used in both imaging applications and dose calculations, and can be run either as a single process or in parallel mode using the EGSnrc parallel script.

2. FEATURES AND COMPARISON

2.1 Input file format

The input file used by the EGSnrc C++ class library consists of key-value pairs in well-defined structure describing all the geometries used in the simulation as well as the particle source and other simulation parameters. To specify a voxelized phantom with many voxels the input files gets rather large and the parsing time becomes very long. It was found that the time increases approximately quadratically with the number of lines.

To reduce the parsing time, the input file is translated into XML and a standard XML parser is used to process it. The XML input file uses exactly the same structure as the original EGSnrc C++ file and contains exactly the same information. In addition, a functionality was implemented to allow references to other files to be placed in the input file. This makes specifying a simulation much more flexible because parts of the input can be saved to a separate file that can be referenced by many other input files.

Although the input file size is essentially the same as that for egspp, the XML input file can be parsed significantly faster (at least 1,000 times) than the original file used by the egspp package as shown in Table 1.

Table 1. Input file size and parsing time comparison between XML and EGSINP formats

phantom size (voxels)	64×64×64	128×128×128	256×256×256	512×512×512
approximate # of lines†	4,000	15,500	61,500	246,000
T_{xml} (ms)	4	16	68	280
T_{egspp}	6 s	96 s	27 min	7.6 h
T_{egspp}/T_{xml}	1,440	6,000	24,000	98,000

† There are three more lines in the XML file than in the egspp file: an XML declaration, along with start and end tag lines for the root element.

Note: T_{xml}: parsing time for the XML input file; T_{egspp}: parsing time for the egsinp input file of the EGSnrc C++ library.

2.2 Running speed

The computational speed of Epp was compared with that of DOSXYZnrc (slightly modified from the official version to output photon information, same for all following refers to it) by running simulations to generate images from primary and scattered x-ray photons. When run in parallel mode, DOSXYZnrc writes several files for each process. When large numbers of histories are needed, the output files will be large to the point of being prohibitively difficult to process with an average computer. As a result, although the node used in our cluster computer has only 8 physical processors, the simulation was run with 20 processes to reduce the output

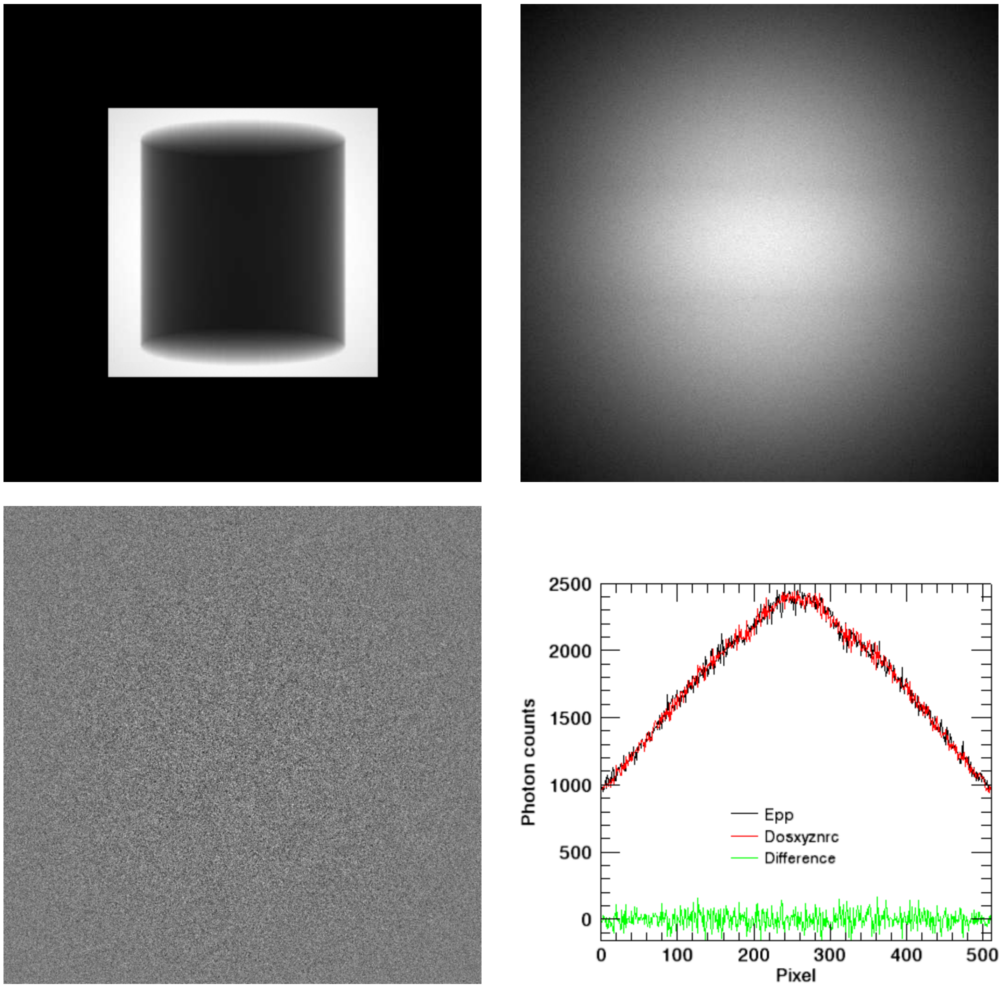

Figure 1. Simulation images of Epp and the comparison with DOSXYZnrc. The two top images are the primary beam and the Compton scatter images, respectively. The bottom left panel is the difference map of the Compton scatter images of the Epp and DOSXYZnrc results. The bottom right figure shows the central row profiles of the Compton scatter images of Epp (black line) and DOSXYZnrc (red line), respectively and the difference (green broken line) between them.

file size. This certainly induces some overhead. However, it should be noted that the result of the performance comparison between the two programs is not affected.

Both programs were run for a 10^{10} photon history simulation in parallel mode with 20 processes on a dedicated cluster node. It takes $\sim$11 hours for DOSXYZnrc to do the simulation, plus $\sim$2 hours for photon propagation with a program written in C. The CPU time for each process is $\sim$4 hours. The total time is $\sim$13 hours for the entire task. The elapsed time for Epp is only $\sim$6.5 hours and CPU time is $\sim$2.2 hours with all the output files saved. Without writing photon information to file, the elapsed time taken by Epp is $\sim$5.6 hours and its CPU time is $\sim$2.2 hours. The time for file I/O is estimated to be $\sim$1 hour. Therefore, Epp is at least twice faster than DOSXYZnrc for an identical task.

2.3 Output file

In our test run, the total size of the photon output files is $\sim$210 GB. Because DOSXYZnrc currently does not perform photon propagation (as it was originally meant for radiation dose calculation), the photon information needed must be saved into a series of output files for propagation later. This will greatly increase the cost from file I/O and data storage. As the speed is significantly improved and the propagation is done on the fly in Epp, it is not necessary to save the photon data to disk when only the projected image is needed. However, if necessary, the resultant photon information can be saved by specifying options with command line arguments. Ideally, the absence of intermediary photon information files will significantly decrease file I/O time due to the relatively low speeds of the external storage devices and network used.

2.4 Simulation results

Epp was compared with DOSXYZnrc by running a simulation to create images by propagating primary and scattered photons to an ideal array detector. The x-ray source used is an ideal monochromatic point source (60 keV) that is collimated to a square at the front surface of the object. the object is a cylindrical breast phantom composed of 50% glandular tissue, 50% adipose tissue, with polyethylene spheres embedded. All the input parameters in both files were the same, and both programs were run on the same node of a Linux cluster with eight 3.16 GHz physical processors and 16 GB shared memory.

In Fig. 1 is an example of the simulation results. The top left panel is the primary image. The top right is the image of Compton scattered photons. The bottom left panel is the difference map of the two Compton scattering images created from the results of DOSXYZnrc and Epp. The bottom right panel is the central row profile of Compton scatter images and the difference map. It is evident from these images that the results are consistent.

3. SUMMARY

Epp is a user code package based on the EGSnrc C++ class library. It can trace particles interacting with matter and is ideally suited for imaging applications and radiation dose calculation. Its functionality can be readily extended by modifying the current code to trace electrons or positrons, as well.

In general, modeling sources and geometries is much easier than in DOSXYZnrc. Using an XML format for the input file makes the file parsing at least 1,000 times faster than the original egspp input file. Epp is at least two times faster than DOSXYZnrc. When only the resultant images are needed, it is not necessary to save the intermediate particle information, resulting in a significant savings of data storage space and time for file I/O.

The package has been validated by comparing the simulation results with that of DOSXYZnrc. Therefore, Epp can be used for the Monte Carlo simulation of particles for radiation dose and imaging applications.

This work has not been submitted for publication or presentation anywhere.

ACKNOWLEDGMENTS

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

REFERENCES

[1] Zhu, L., Wang, J., and Xing, L., "Noise suppression in scatter correction for cone-beam CT," *Medical Physics* **36**(3), 741–752 (2009).

[2] Bhagtani, R. and Schmidt, T. G., "Simulated scatter performance of an inverse-geometry dedicated breast CT system," *Medical Physics* **36**(3), 788–796 (2009).

[3] Chen, Y., Liu, B., O'Connor, J. M., Didier, C. S., and Glick, S. J., "Characterization of scatter in cone-beam ct breast imaging: Comparison of experimental measurements and monte carlo simulation," *Medical Physics* **36**(3), 857–869 (2009).

[4] Zhu, L., Xie, Y., Wang, J., and Xing, L., "Scatter correction for cone-beam CT in radiation therapy," *Medical Physics* **36**(6), 2258–2268 (2009).

[5] Walters, B., Kawrakow, I., and Rogers, D., *DOSXYZnrc Users Manual*. National Research Council of Canada, Ottawa K1A 0R6 (July 2009).

[6] I. Kawrakow, E. Mainegra-Hing, F. T. and Walters, B., *The EGSnrc C++ class library, NRC Report PIRS-898 (rev A)*. National Research Council of Canada, Ottawa, K1A OR6 (July 2009).