POLY2TET: A Program to Convert Polygonal Surface Mesh Phantom into Tetrahedral Volumetric Mesh Phantom

Haegin Han [a*], Chansoo Choi [a], Bangho Shin [a], Yeon Soo Yeom [b], Chan Hyeong Kim [a*]
[a]*Department of Nuclear Engineering, Hanyang University, 222 Wangsimni-ro, Seongdong-gu, Seoul 04763, Korea*
[b]*National Cancer Institute, National Institute of Health, 9609 Medical Center Drive, Bethesda, MD 20850, USA*
[*]*Corresponding author: chkim@hanyang.ac.kr*

1. Introduction

For the last two decades, hundreds of BREP (boundary representation) computational phantoms have been developed, recognized as the third generation of phantoms which overcame various limitations of the second-generation voxel phantoms [1]. These BREP phantoms, mostly composed of NURBS (Non-Uniform Rational B-Splines) and polygonal mesh (PM) surfaces, are featured by their flexibility that can define even small or thin structures and high deformability that enables the phantom deformation into different body sizes or postures [2-4]. However, except for Geant4 [5] where PM geometry can be implemented by *G4TessellatedSolid* class, currently there is no Monte-Carlo particle transport code in general purpose (e.g., MCNP6, PHITS, and FLUKA) where NURBS or PM geometries can be directly implemented. Therefore, to be used for the dose calculations, most of the BREP phantoms should be voxelized at user-defined resolutions (i.e., typically 0.1-few millimeters), which leads to the conventional limitations of voxel phantoms (e.g., stair-step surface, voxel effect, and incapability to define thin layers).

The need of voxelization procedure and the consequent limitations could be recently obviated by the emergence of tetrahedral mesh (TM) phantom suggested by Yeom *et al.* [6]. TM phantoms can be directly generated from PM phantoms while preserving the exact contours defined by polygon surfaces, and also can be generated from NURBS phantoms after the conversion into PM phantoms by using 3D programs (e.g., Rhinoceros 3D, Maya, and Blender). Unlike BREP phantoms, TM phantoms can be directly implemented into the wider range of Monte-Carlo codes, including MCNP6, PHITS, and Geant4. Moreover, it was demonstrated that TM format could be more advantageous than PM format in terms of computational speed, since it has only four facets for each element while a single polygon typically consists of at least hundreds of facets [6]. Acknowledging such advantages, the International Committee on Radiological Protection (ICRP) recently converted voxel-type ICRP-110 reference phantoms into TM format [2], and ICRP Publication on the newly developed mesh-type reference computational phantoms (MRCPs) will be released by the end of this year. The ICRP Publication, which will be provided with MRCPs themselves (both in PM and TM format) and sample codes to implement TM phantom into MCNP6, PHITS, and Geant4, may stimulate the development of various TM phantoms.

In the present study, to facilitate the tetrahedralization of PM phantom in *.obj* file into TM phantoms (*.ele* and *.node* files), a comprehensive C++ program – POLY2TET – was developed. The program was developed by using the source code of TetGen 1.5.1 [7] and includes required preprocessing and postprocessing procedures. The developed program was verified by converting three different adult male PM phantoms into TM format and comparing the organ masses of the converted TM phantoms with those of the original PM phantoms. The compatibility of TM phantoms with Geant4, MCNP6, and PHITS was also investigated by using those three codes to calculate dose coefficients (DCs) in a generated TM phantom for broad photon and electron beams in AP-direction. In addition, to estimate computation performance in Geant4 for both PM and TM phantoms, the memory consumption and computational time were measured for the implementation of three different phantoms.

2. Material and Methods

2.1 POLY2TET

POLY2TET is a program to convert PM phantoms in *.obj* file into TM phantoms in *.ele* and *.node* files. The group names in *.obj* file need to start with an organ ID (a positive integer), and underscore (_) should be added before any additional word. The source code of POLY2TET was written by using the source code of TetGen 1.5.1, which is a program to generate tetrahedral meshes in 3D domains defined by polygonal surfaces. As shown in Figure 1, the overall POLY2TET program consists of preprocessing, tetrahedralization, and postprocessing procedures.

Preprocessing procedure is where the *.obj* file is converted into supported input formats for TetGen (*.node* and *.smesh* files). For this, the shell groups defined in PM phantom (*.obj* file) are first split into single polygons to establish all the 3D domains. Please note that each of the polygons defines each domain. Then, an arbitrary point for each domain is calculated by moving a vertex on each polygon 0.001 μm inside. By using the calculated points and vertex/facet information given in *.obj* file, *.smesh* and *.node* files are generated.

In tetrahedralization procedure, TetGen is called with the option *-p/0.001YAFT1e-9*. During the tetrahedralization, when intersected mesh is detected,

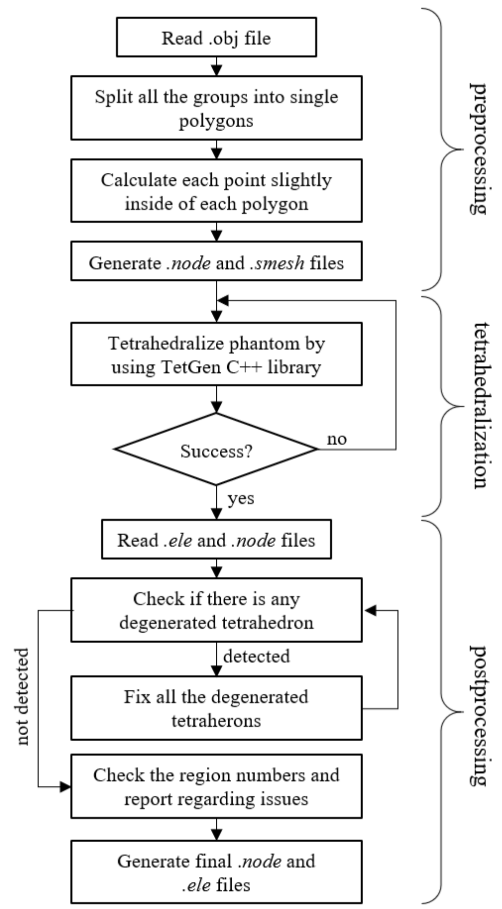

Fig. 1. Procedure flowchart of POLY2TET.

TetGen reports the error message with the intersected position. However, when TetGen fails to generate tetrahedral mesh even without any intersected part, which hardly occur, TetGen is repeatedly called until the tetrahedralization succeeds. When it succeeds, TetGen generates *.node* and *.ele* file for tetrahedral mesh, which contains the vertex and element (tetrahedron) information, respectively. The *.ele* file also contains region number for each tetrahedron, which is given by the organ ID defined in PM phantom (*.obj* file).

Postprocessing part is in charge of the modification of degenerated tetrahedrons and wrong region numbers. It first reads newly generated *.node* and *.ele* files and checks if there is any degenerated tetrahedron according to the definition of degenerated tetrahedron given in the source code of G4Tet class. Note that those degenerated tetrahedrons are not allowed to be implemented in Geant4 by default. The degenerated tetrahedrons are then fixed by repeatedly moving a vertex of the tetrahedron at a very short distance (<0.001 μm). The process to check and fix degenerated tetrahedrons is iterated until none of them are left. Finally, the region numbers are investigated if there is any number not defined by organ ID. Among those wrong region numbers, the zeroes, rarely generated due to the bug in TetGen code, are replaced by the organ ID with the largest volume (or user-defined ID given with option -c). For the non-zero region numbers, which can be generated due to the polygons with holes or inversed normals, the location of the corresponding tetrahedron is reported, so that user can fix the defects in PM phantom.

2.2 Software Verifications

To verify POLY2TET, three different adult male phantoms (i.e., MRCP; simplified MRCP, without target layers and blood vessels; and MRKP [8]) in PM format were converted into TM format. Then, to investigate the compatibility of TM phantom with Geant4, MCNP6, and PHITS, a TM phantom (MCRP) was implemented into the codes to calculate skin and liver DCs to broad parallel beam with 0.01-100 MeV photon/electron energies. Note that skin doses were calculated in skin basal cell layer defined in a depth of 50-100 μm. The number of primary particles was chosen to keep the statistical relative error of the DC values below 5%. The calculated DCs for skin and liver were then compared with those calculated from PM phantom implemented in Geant4.

The comparison on memory consumption and computational time was made between PM and TM phantoms. The measurements were conducted only by using Geant4, since PM phantom cannot be implemented in MCNP6 and PHITS. Computational time was measured for initialization process and particle transport. The source particles were set as photons and electrons with the energies between 0.01 and 100 MeV and were generated uniformly in the whole body of the phantoms. The simulations were performed on a single core of the Intel® Xeon® Platinum 8160 CPU (@ 2.10 GHz CPU processor and 256 GB RAM memory)

3. Result

The conversion procedure by POLY2TET was very stable for all the three phantoms (i.e., MRCP, simplified MRCP, and MRKP), taking less than five minutes of computational time. Figure 1, as an example, shows adult male MRCP in PM format and that in TM format generated by POLY2TET. It can be seen that TM was generated while well-conserving the contours defined by PM geometries. The organ volumes were estimated in PM and TM phantoms, and it was confirmed that the differences in organ volume between PM and TM formats are less than 0.001%. It should be noted that TetGen did not generate any degenerated tetrahedron during the conversion of these three phantoms, so the function to modify degenerated tetrahedrons was separately verified by using extremely thin phantoms. For all the phantoms, the volume with the zero-region number was less than 0.5 cm^3, which were assigned with the organ ID of residual tissue.

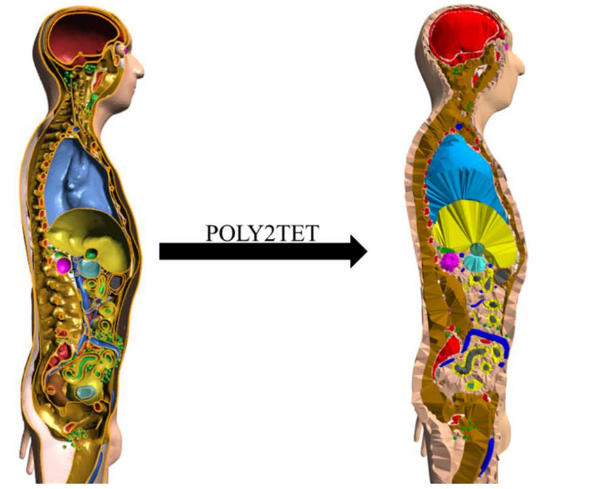

Fig. 3. Adult male MRCP in PM format (left) and that converted into TM format (right) by POLY2TET.

Table1. Skin (basal cell layer) and liver DCs (dose coefficients) calculated with adult male MRCP in PM and TM format by using Geant4, MCNP6, and PHITS. (unit: pGy·cm^2)

		PM		TM					
		Geant4		Geant4		MCNP6		PHITS	
	(MeV)	Skin	Liver	Skin	Liver	Skin	Liver	Skin	Liver
Photon	0.01	3.3	6.0E-6	3.3	5.9E-6	3.3	6.6E-5	3.3	5.7E-5
	0.1	0.4	0.5	0.4	0.5	0.4	0.5	0.4	0.5
	1	2.5	4.1	2.5	4.1	2.4	4.1	2.5	4.2
	10	9.7	22.5	9.7	22.4	10.5	22.3	9.9	22.6
	100	38.2	76.5	38.2	75.7	40.0	76.9	39.2	76.6
Electron	0.01	0	0	0	0	0	0	0	0
	0.1	306.7	9.0E-5	306.7	9.1E-5	312.3	9.3E-5	314.1	7.5E-5
	1	182.3	1.2E-2	182.3	1.2E-2	184.5	1.3E-2	181.2	1.2E-2
	10	187.4	56.3	187.4	56.6	201.2	56.1	185.7	58.4
	100	310.5	354.5	310.5	349.3	314.8	350.2	309.1	353.6

Table 1 compares the skin and liver DCs calculated with adult male MRCP in TM format with those calculated with the original PM phantom. It can be seen that the differences in DC values of TM and PM phantoms are less than 3% for all cases, verifying the good agreement between those two phantoms. DCs of TM phantoms calculated by MCNP6 and PHITS showed larger differences from those of PM phantoms, which may be due to the code differences, but still they were all less than 17%.

Fig. 4 compares memory usage by PM and TM phantoms during particle transport in Geant4 for three different adult male phantoms. TM phantoms consumed ~6 times more memory than corresponding PM phantoms. However, all TM phantoms required less than 8GB memory which can be provided by typical personal computers. Computational time required for initialization of TM phantoms in Geant4 was generally 5-7 min, while it was 20-40 min for PM phantoms. Fig. 5 compares the computation time in Geant4 to transport 10^5 particles uniformly generated in PM and TM phantoms for three different phantoms. It can be seen that computational time was shorter with TM phantoms for all cases, showing up to ~25 faster speed than PM phantoms.

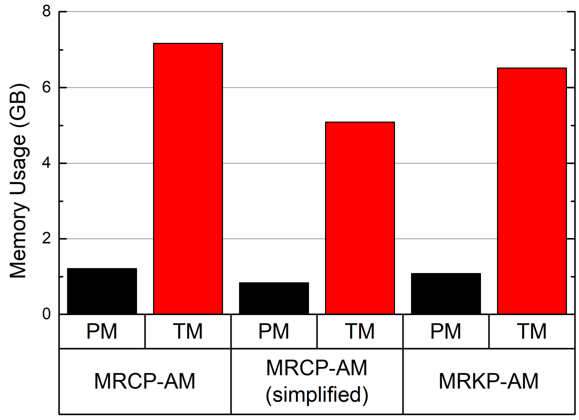

Fig. 4. Memory usage during the particle transport in Geant4, when three different adult male phantoms in PM and TM format are implemented.

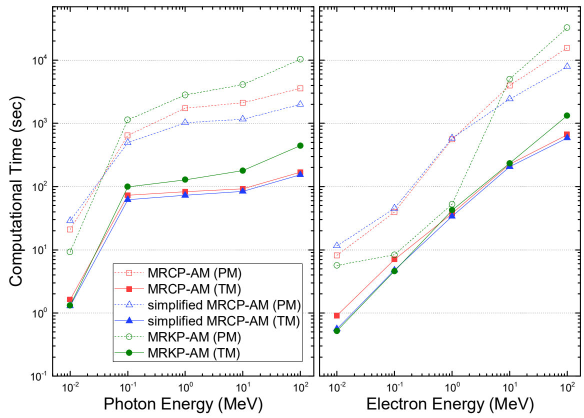

Fig. 5. Computational time in Geant4 to transport 10^5 photon and electron primary particles uniformly generated in the whole body of three different phantoms in PM and TM format.

4. Conclusions

In the present study, POLY2TET program was developed to facilitate the conversion of PM phantom (*.obj* file) into TM phantom (*.ele* and *.node* files). Based on TetGen C++ library, POLY2TET consists of three procedures. These procedures were verified by converting some PM phantoms into TM format. The compatibility of TM phantoms with Geant4, MCNP, and PHITS was also confirmed by comparing DCs for TM phantoms calculated by those three codes with those for PM phantoms calculated by Geant4. Regarding computation performance in Geant4, initialization and particle transport speed was up to ~6 and ~25 times faster for TM phantoms, respectively, while TM phantoms consumed ~6 times more memory. However, the memory required for the implementation of TM phantom in Geant4 was less than 8GB, which can be provided by typical personal computers. POLY2TET is expected to be a useful tool to convert various BREP phantoms into TM phantoms, substituting conventional voxelization process. In the future, the competitiveness of POLY2TET will be further investigated by comparing the performance of the TM format with that of DagMC

[9] which is a plug-n software that allows users to implement CAD geometry into various Monte-Carlo codes (e.g., MCNP5, MCNP6, and FLUKA). The POLY2TET program with its source code can be distributed upon request.

REFERENCES

[1] X. G. Xu, An exponential growth of computational phantom research in radiation protection, imaging, and radiotherapy: a review of the fifty-year history, Physics in Medicine & Biology, 59(18), 2014.

[2] C. H. Kim, Y. S. Yeom, T. T. Nguyen, et al., New mesh-type phantoms and their dosimetric applications, including emergencies, Annals of the ICRP, 47(3-4), p. 45-62, 2018.

[3] A. M. Geyer, S. O'Reilly, C. Lee, et al., The UF/NCI family of hybrid computational phantoms representing the current US population of male and female children, adolescents, and adults—application to CT dosimetry, Physics in Medicine & Biology, 59(18), p. 5225, 2014.

[4] P. A. Lombardo, F. Vanhavere, A. L. Lebacq, et al., Development and Validation of the Realistic Anthropomorphic Flexible (RAF) Phantom, Health physics, 114(5), p. 486-499, 2018.

[5] J. Allison, K. Amako, J. Apostolakis, P. Arce, et al., Recent developments in Geant4. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 835, p. 186-225, 2006.

[6] Y. S. Yeom, J. H. Jeong, M. C. Han, et al., Tetrahedral-mesh-based computational human phantom for fast Monte Carlo dose calculations, Physics in Medicine & Biology, 59(12), p. 3173, 2014.

[7] H. Si, TetGen, a Delaunay-based quality tetrahedral mesh generator, ACM Transactions on Mathematical Software (TOMS), 41(2), p. 11, 2015.

[8] C. Choi, T. T. Nguyen, Y. S. Yeom, H. Lee, et al., Mesh-type reference Korean phantoms (MRKPs) for adult male and female for use in radiation protection dosimetry, Physics in Medicine & Biology, 64(8), 085020, 2019.

[9] T. J. Tautges, P. P. H. Wilson, J. Kraftcheck, et al., Acceleration techniques for direct use of CAD-based geometries in Monte Carlo radiation transport, in: International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2009), American Nuclear Society, Saratoga Springs, NY, 2009.