RPE | Radiation Physics & Engineering
A Peer-Reviewed Journal
Published by K.N. Toosi University of Technology
http://rpe.kntu.ac.ir
Research Article

Radiation Physics and Engineering 2024; 5(2):61–69

A comprehensive approach for calculation of surface area and volume in radiation transport codes

Hamed Kargaran*
Faculty of Engineering, Shahed University, P.O. Box 33191-18651, Tehran, Iran

HIGHLIGHTS

- The proposed method can compute all types of surface areas and volumes for complex geometries.
- Reasonable accuracy and precision were obtained in a relatively short computational time.
- The presented modules can be used effectively in any Monte Carlo-based code that employs CG geometry modelling.

ABSTRACT

Abstract: The estimation of flux in radiation transport Monte Carlo problems needs to calculate the volumes and surface areas of the geometric regions. The particle flux is often estimated as the track length per unit volume or the number of particles crossing a surface per unit area in Monte Carlo transport problems. Various representations such as constructive solid geometry (CSG), boundary representation (B-Rep), and combinatorial geometry (CG) are proposed in the literature for geometry modeling and calculation of surface area and volume. MCNP series and OpenMC as Monte Carlo particle transport codes utilize CG modeling and are not able to calculate surface area as well as volume for non-rotationally symmetric or non-polyhedral cells. In this work, a comprehensive approach based on the Cauchy-Crofton formula using the Monte Carlo method has been implemented to the radiation transport codes as an extra module for computing surface area and volume of complex geometries. We used a random sampling procedure to create the required probe lines and points in the computational approach. The results show that this method can accurately compute surface areas and volumes of complex geometries with a relative error of less than 0.1% and a short computation time of a few seconds, which is not achievable with the cuurent MCNP and OpenMC modules.

KEYWORDS

Monte Carlo
Cauchy-Crofton formula
Surface area
Volume
MCNP
OpenMC

HISTORY

Received: 26 September 2023
Revised: 27 February 2024
Accepted: 4 March 2024
Published: Spring 2024

1 Introduction

Objects' geometric properties, such as length, area, and volume, are significant quantities that need to be frequently calculated in many biological, medical, and industrial applications (Timmer and Stern, 1980). For instance, the cortical surface area can be concerned with functional capacities in the analysis of the cortex in MR images (Windreich et al., 2003; Zeng et al., 1999). Furthermore, the peritoneal surface area is considered an important factor in dialysis effectiveness in hemodialysis (Breton et al., 2008). 3D shape recognition and matching which focus on the area of geometric measurements can also be another class of applications (Klette and Rosenfeld, 2004). On the other hand, the radiation flux in Monte Carlo transport problems is often estimated as the track length per unit volume or the number of particles crossing a surface per unit area. Therefore, knowing the volumes and surface areas of the geometric regions in a Monte Carlo problem is essential (Hendricks, 1980). Knowing volumes is useful in calculating the masses and densities of cells and thus in calculating volumetric or mass heating. Moreover, the calculation of geometrical mass is frequently a good check on the accuracy of the geometry setup when the mass is known by other means.

There are different representations with various approaches currently in use for calculating the area, volume, and other measures of geometric objects (Breton et al., 2008; Castro and Sbert, 2000; Dorst and Smeulders, 1987; Flin et al., 2005; Gärtner, 1999; Klette and Sun, 2001; Sarraga, 1982; Fang et al., 2009). One approach to calcu-

*Corresponding author: kargaran@shahed.ac.ir

https://doi.org/10.22034/rpe.2024.418185.1164

lating volume properties reduces them to the calculation of appropriate surface integrals by means of the divergence theorem (Klette and Sun, 2001). Thus, calculating integrals over the surface of the object can be applied to compute other mass properties of interest.

Another approach, used in GMSolid, is described by Sarraga as an alternative means of evaluation (Sarraga, 1982). In this method, the surface of each primitive is divided into small elements, whose sizes are chosen to meet a user-defined density. These elements are topologically rectangular and are bound by parameter lines. Each element is then classified as lying inside, outside, or on the model by means of point membership classification, using a random test point belonging to the element. The area of each element classified as on the surface of the solid is computed analytically using the usual formula from differential geometry, which is represented by Eq. (1):

$$A = \int_{\nu_1}^{\nu_2} \int_{u_1}^{u_2} \sqrt{(r_u.r_u)(r_\nu.r_\nu) - (r_u.r_\nu)^2}\, \mathrm{d}u\mathrm{d}\nu \qquad (1)$$

where the parameter lines are expressed by $u = u_1$, $u = u_2$ and $\nu = \nu_1$; $\nu = \nu_2$. These areas are then summed to give an estimate for the surface area of the whole object. In another class of approaches, which is based on ray casting, the area of a surface can be approximated as the sum of the areas of rectangular strips (Roth, 1982). Each surface is covered by a bounding rectangle, which is divided into strips. Within each strip, a ray is produced across the rectangle. Then, the location of the entrance and exit through the surface is recorded. An approximation for the area of the surface is given by summing the lengths of the segments inside the surface which is multiplied by the widths of each strip. In addition, a regular grid of rays can be shot through the object and then recorded as entry and exit points. Adjacent piercing points are connected to form triangles, and the sum of the areas of these triangles approximates the surface area of the object (Prisant, 1996). Clearly, this is not accurate where the rays are almost tangential to the object.

In addition, boundary representation (B-Rep) modeling can be used to represent the geometries as another approach. This method allows producing a polygonal tessellation of the surface of the model (Lorensen and Cline, 1998). Summing the areas of all polygons in the tessellation can obtain an approximation of the surface area.

The probabilistic method of quasi-Monte Carlo area (QMCA) was presented by Liu for computing the surface area of a complicated geometry as the point-sampled surfaces (Liu et al., 2006) in CSG modeling. It is based on the Cauchy-Crofton formula (Do Carmo, 2016) and performs the area estimation by counting the number of intersection points between the point cloud and a set of uniformly distributed lines generated with low discrepancy sequences. In essence, the Cauchy-Crofton formula transforms a problem of estimating the surface area into a problem of counting the intersection points. The algorithm of a line and point sets intersecting (LPSI) was applied to determine all intersection points between the point set and the lines, which is based on a clustering technique. In some sense, the LPSI algorithm can be considered an extension of the ray tracing routine proposed by Schaufler et al. (Schaufler and Jensen, 2000). As reported in (Liu et al., 2006), the results of point classification in this algorithm are dependent on the orientation of the normal vector, which may lead to large approximation errors. Therefore, converting geometry to point-sampled surfaces is an extra stage. In addition, using this method will inherently lead to a geometric approximation.

The estimation of the particle flux by Monte Carlo codes, such as MCNP or OpenMC, requires in advance the calculation of volumes and surface areas of the geometric regions of interest (Brown et al., 2002; Romano et al., 2015; Romano and Forget, 2013). These are general-purpose Monte Carlo codes, that use combinatorial geometry modeling (CG). Calculating volumes and surface areas in modern Monte Carlo transport codes is difficult because MCNP as well as OpenMC allows the construction of cells from unions and/or intersections of regions defined by an arbitrary combination of second-degree surfaces, toroidal fourth-degree surfaces, or both. These surfaces can have different orientations or be segmented for tallying purposes. Also, the cells can be constructed from quadrilateral or hexagonal lattices or can be embedded in repeated structures universes. Although such generality greatly increases the flexibility of MCNP, computing cell volumes, and surface areas understandably requires increasingly elaborate computational methods.

MCNP cannot calculate the volumes and areas of asymmetric, non-polyhedral, or infinite cells and in these cases, the volume and area calculation can fail because of round-off errors (Hendricks, 1980; Brown et al., 2002; Romano et al., 2015; Brown et al., 2003). However, MCNP uses a stochastic procedure that can estimate area and volume using the ray-tracing method (Goorley et al., 2014). This procedure requires changing the input and executing the code separately. The cell flux tally and surface flux tally are inversely proportional to cell volume and area, respectively. The input for the main run will then contain the calculated area and volume. A thorough Monte Carlo method has been added as a module to the MCNP series to efficiently compute the surface area and volume of complex geometries in CG modeling without the need for repeated executions of the MCNP code. This module accurately computes the surface area and volume of geometries within a reasonable computation time.

2 Materials and Methods

2.1 Area calculation

To calculate surface area in this approach, inspired by the QMCA method, we implemented a similar Monte Carlo method in the CG representation. A sphere surrounding is considered on the desired geometry. A sufficient number of uniformly distributed lines are defined to cross the sphere and geometry surfaces randomly. Since boundary surface equations are known in the CG method, unlike the LPSI algorithm in the QMCA method, no extra approximation is performed in calculating the surface area due to converting geometry to point-sampled surfaces. Surface

area estimation can be carried out by the Monte Carlo method according to Eq. (2) (Liu et al., 2006):

$$S \approx \frac{n}{n_1} S_1 \tag{2}$$

where S_1 is the surface of a circumscribing sphere, S is the surface of the desired geometry, n_1 is the total number of intersection points concerning probe lines crossing the sphere and n is the total number of intersection points concerning probe lines crossing the desired geometry.

For simple geometries, that is those defined by primitive surfaces only and not involving further combinatorial operations, finding the number of intersection points on the surfaces of the desired geometry is simple. In complex geometries due to the union and intersection of multivolumes, a failure may be revealed to compute the surface area. The source of error in this case is due to considering those intersection points, corresponding to surfaces between unified cells, which are not included as the surface border points.

To resolve this problem, at the intersection point of probe lines with surfaces, two adjacent points around each intersection point are selected as shown in Fig. 1. If all of these three points are located in the same cell, they correspond to surfaces between unified cells and are not considered in the computational approach. Conversely, the intersection point will be considered as a surface border point if one of the two adjacent points is located in a different cell.

An illustrative example of the surface border point selection method is depicted in Fig. 1. As is shown, to distinguish the intersection points on the surface "S" as the border points or not, the two probe lines A and B are crossed with that surface. Based on the proposed approach both adjacent points in crossing point (a) are located in the same cell. Therefore, this point is not considered in the surface area calculation for cell 1. On the other hand, as one of the adjacent points in crossing point (b) is out of cell 1, consequently point (b) is considered as a border point in the calculation.

The above-mentioned approach was implemented as an extra module in MCNP code to estimate the surface areas that could not be computed by using current versions of the code. The algorithm of the approach is shown in Fig. 2. Where m is the number of intersection points on the investigated surface per each line and n is the total trial number of probe lines sampling. For each cell in the algorithm, the parameters associated with the limiting planes are loaded. According to the specified procedure, verifying if these points are part of the cell is conducted. Finally, the surface area is computed using Eq. (2). This approach has been implemented as a Monte Carlo Surface Calculation (MCSC) module in MCNP.

Due to the probabilistic nature of this approach, the precision of the calculated area increases with probe line sampling. It should be noted that the determination of the intersection point belonging to the cell has been carried out by the CHKCEL subroutine in MCNP (Hendricks, 1980).

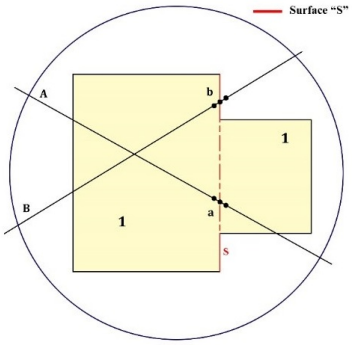


Figure 1: Surface border points selection approach by using the Monte Carlo method.

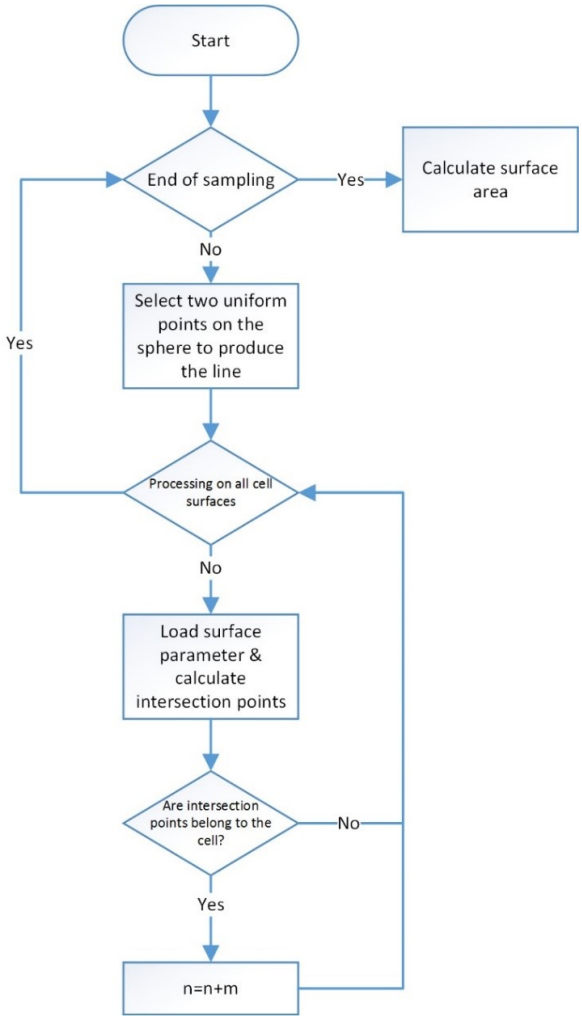


Figure 2: The implementation approach for estimating the surface area in MCNP.

2.2 *Volume calculation*

To calculate the volume of a desired cell in this approach, the Pi estimation idea by using the Monte Carlo method has been utilized (McCloskey and Braithwaite, 1995; Kargaran et al., 2016). Then, the sampling of random points is considered within a cube surrounding the desired cell. The volume of the desired cell is then calculated according

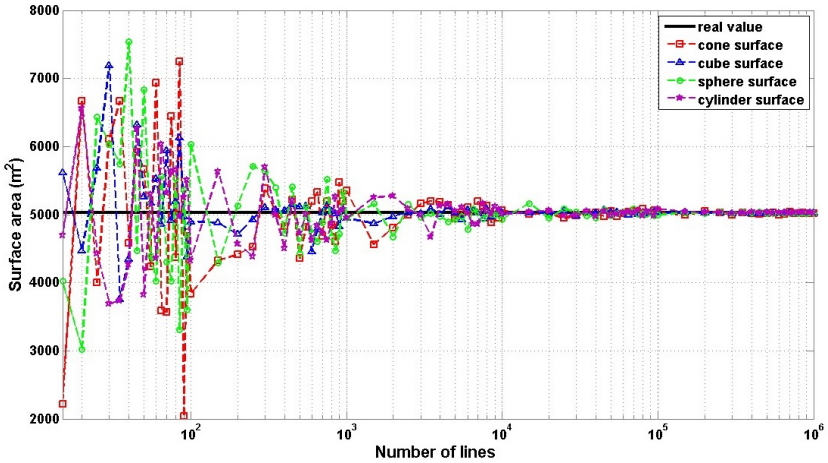


Figure 3: Sensitivity analysis for estimating the area of four objects with a constant surface area.

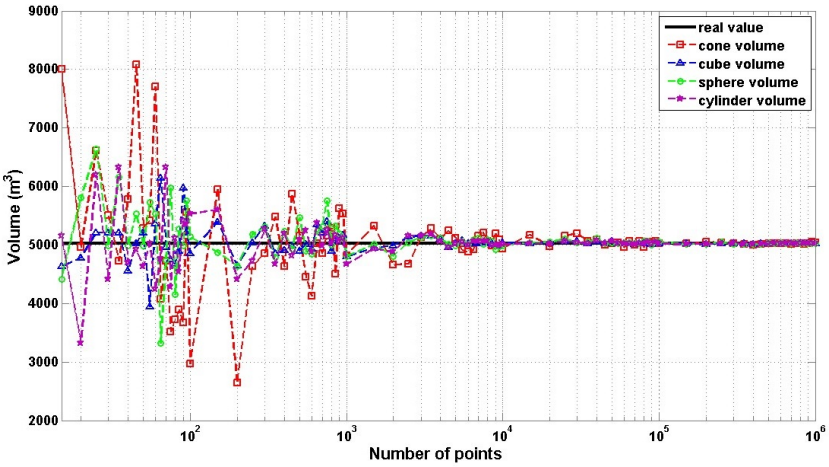


Figure 4: Sensitivity analysis for estimating the volume of four objects with a constant volume.

to Eq. (3):

$$\frac{\text{Desired volume}}{\text{Cube volume}} = \frac{N_{\text{hit}}}{N_{\text{total}}} \tag{3}$$

where N_{total} is the total number of sampling points in the inscribed cub and N_{hit} is the total number of those sampling points that fall within the desired geometry. Similar to the MCSC method precision of the calculated volume increases with point sampling. This approach has been implemented as a Monte Carlo Volume Calculation (MCVC) module in MCNP. As illustrated in section 2.1, the definition of a point belonging to the cell has been done by the CHKCEL subroutine.

3 Results and Discussion

The algorithms presented here are developed and compiled in the FORTRAN programming language. All the simulations were run on a PC with Intel Core i7-4790 3.60 GHz CPU and 16 GB RAM.

To evaluate the performance of surface area estimation, we considered four simple objects (sphere, cube, cylinder, and cone) with the same surface area of 5026.55 m^2 as illustrative examples. Figure 3 shows the curves of sensitivity analysis corresponding to the required probe lines for estimating the area of the objects, where the number of lines is specified from 10 to 1,000,000. It can be seen that the presented method leads to reaching the real values with small approximation errors with an increasing number of lines. When the number of lines was equal to 30,000 the relative error was approximately 1%. To obtain a relative error of less than 0.5%, the number of random probe lines should be 200,000 at least.

In addition, the performance of the presented method was evaluated to estimate the volume by testing simple objects with the same volume of 5026.55 m^3. The curves of sensitivity analysis corresponding to the required points for estimating the volume of four simple objects are shown in Fig. 4. As indicated, increasing the number of points can also lead to small approximation errors. The relative error of 1% was achieved when the number of points was equal to 35,000. Moreover, the number of random points above 150,000 caused a relative error of less than 0.5%.

The mean and standard deviation for a population of 200 times volume and surface area calculations, using 1,000,000 points and probe lines for all of the geometrical

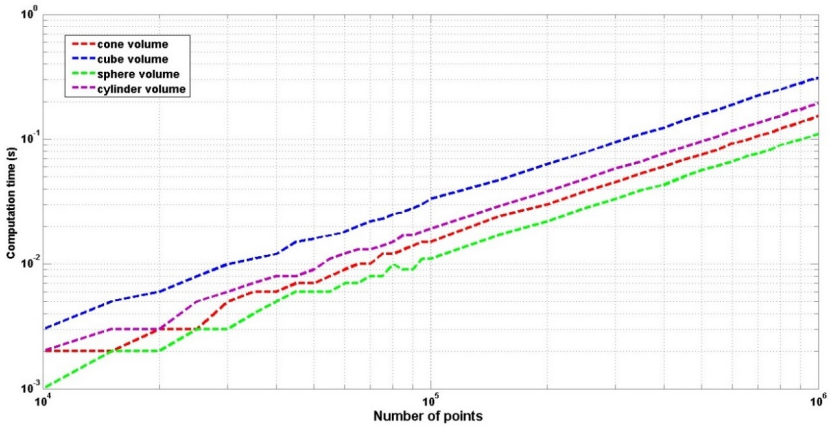


Figure 5: The surface area computation time as a function of the number of probe lines.

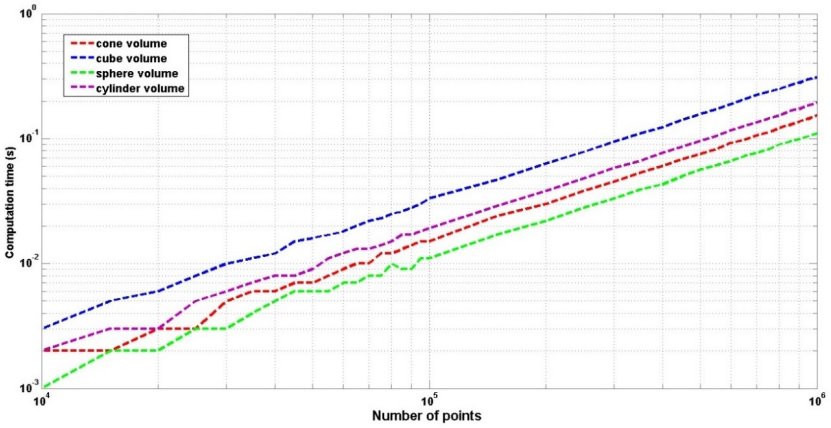


Figure 6: The volume computation time as a function of the number of sampling points.

Table 1: Surface area calculations using the MCSC method in simple geometries.

Geometric object	Analytical calculation (cm^2)	presented module calculation			
		Average (cm^2)	STD.	Relative error (%)	Time (s)
Cone	5026.548	5019.837	10.4228	0.13352	1.081765
Cube	5026.548	5026.582	4.575936	0.00068	3.429615
Sphere	5026.548	5026.12	8.858824	0.00852	0.525745
Cylinder	5026.548	5026.5	7.623056	0.00095	1.6511

Table 2: Volume calculations using the MCVC method in simple geometries.

Geometric object	Analytical calculation (cm^2)	presented module calculation			
		Average (cm^2)	STD.	Relative error (%)	Time (s)
Cone	5026.548	5026.4061	8.162198	0.00282	0.208925
Cube	5026.548	5026.5332	2.81182	0.00030	0.342925
Sphere	5026.548	5026.5172	5.162839	0.00061	0.125865
Cylinder	5026.548	5026.5177	4.479972	0.00060	0.216425

samples, are presented in Table. 1 and Table. 2 respectively. The results indicate that the computation time for both MCSC and MCVC methods rises in direct proportion to the increase in bounding surfaces. The sphere with one surface has the shortest computation time, whereas the cube with 6 bounding surfaces has the longest. The cube has the lowest relative inaccuracy, whereas the cone has the largest.

The computation time versus the number of probe lines and points corresponding to surface area and volume estimation are shown in Figs. 5 and 6, respectively. The results indicate that the computation time increases with the number of lines and points linearly. As depicted in Fig. 5, the computation time is proportional to the degree and the number of constitutive geometrical surfaces. The computation time for cylinder volume calculation, which has two first-degree and a second-degree surface is about 1.75 times more than that of sphere volume. This time

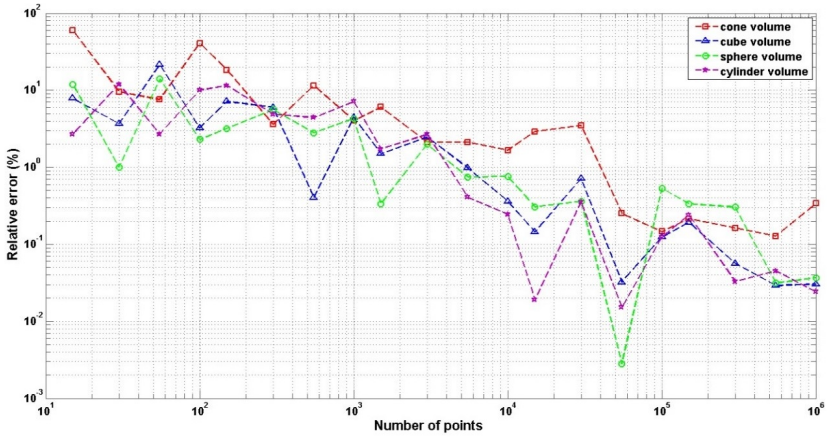


Figure 7: The relative error of volume estimation results versus the number of points.

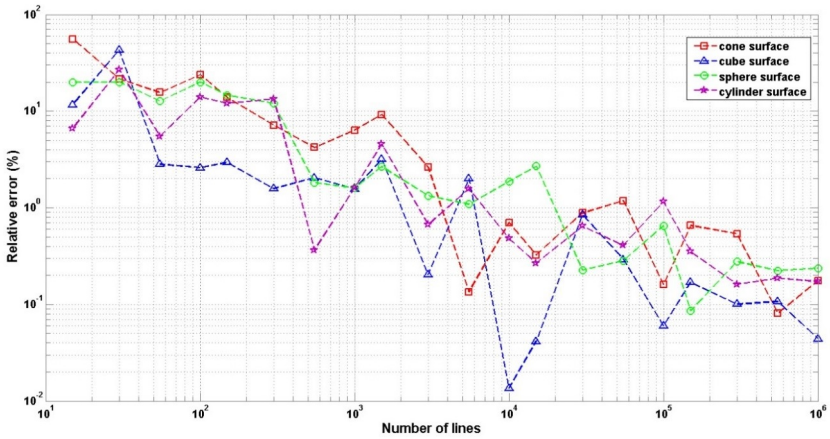


Figure 8: The relative error of surface area estimation results versus the number of probe lines.

Table 3: The detailed analysis concerned with the surface area calculation of sample A.

Surface	Analytical calculation of area (cm^2)	Presented module calculation		
		Average	STD	Relative error (%)
A1	5	5.001639	0.015463	0.032776
A2	5	5.001807	0.015629	0.036146
A3	5	4.99905	0.01644	0.018995
A4	5	5.000589	0.015244	0.011773
A5	1.6438	1.644069	0.008557	0.016358
A6	4	3.998851	0.014709	0.028716
A7	9.4248	9.423864	0.029928	0.009931
total	35.0686	35.06987	0.046597	0.00362

for cone and cube volumes is approximately 1.4 and 2.8 times more than that of the sphere. The maximum computation time is related to the cube geometrical volume which is composed of the six first-degree surfaces and is about 2.8 times of sphere.

To the presented algorithm, the computation time for estimating the surface area has generally revealed a considerable change as compared to the volume calculation for the same geometry.

In the surface area calculation, the elapsed time to compute the intersection point of the probe line with each surface has been considered as well as the degree and the number of constitutive geometrical surfaces. Therefore, the number of geometrical surfaces appears as the most significant factor determining the time of calculations. As shown in Fig. 6, the computation time for the geometry of a sphere, cone, cylinder, and cube using 1,000,000 random probe lines is 0.441, 1, 1.49, and 3.52 s, respectively.

The relative errors between the real value and the results of our approach for estimating of surface area and volume are shown in Figs. 7 and 8, respectively.

It can be seen that the errors decreased with the number of sampling points. The relative error was about 10% at random points of 1,000 and approximately less than 0.07% at 1,000,000 points in volume calculations. In addition, at 1,000,000 sampling points, the maximum relative

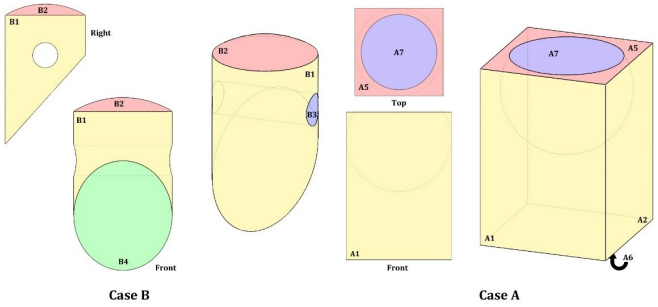


Figure 9: The two samples with complex geometries for validating test.

Table 4: The detailed analysis concerned with the surface areas calculation of sample B.

Surface	Analytical calculation of area (cm^2)	Presented module calculation		
		Average	STD	Relative error (%)
B1	189.7526	189.9655	0.583675	0.112181
B2	88.21613	88.21554	0.367514	0.000666
B3	62.07802	62.20824	0.410714	0.209769
B4	111.0721	111.0467	0.424418	0.022826
total	451.1189	451.436	0.908052	0.070302

Table 5: The detailed analysis concerned with the volume calculations of each sample.

Sample object	Analytical calculation of volume (cm^3)	Presented module calculation		
		Average	STD	Max. Relative error (%)
A	6.4657	6.465939	0.008431	0.134
B	531.1295	531.0816	1.027074	0.202

error is about 0.16% corresponding to the cone geometry as well as a minimum relative error of 0.05% for the cube geometry. Therefore, according to the results, the volume calculation accuracy will be improved by decreasing the degree of surface curvature.

Furthermore, the relative error in surface area calculations was about 10% at random probe lines of 1,000 and approximately less than 0.1% at 1,000,000 lines. The maximum relative error is about 0.34% corresponding to the cone geometry as well as a minimum relative error of 0.09% for the cube geometry. Therefore, according to the results, the area calculation accuracy will be improved by decreasing the degree of surface curvature (from second-degree to first-degree surfaces) for the same number of probe lines.

Here, the two samples with complex geometries in which their volumes and surface areas cannot be calculated by MCNP code were considered as shown in Fig. 9. However, they can be calculated analytically, the designed Monte Carlo approach was used as an extra module to calculate the volumes and surface areas. The detailed analysis of the mean values, standard deviations, and relative errors corresponding to the surface area calculation of each sample (A and B) are summarized in Tables 3 and 4.

The results show that the maximum relative errors between the mean value and analytical calculation were about 0.036% and 0.209% corresponding to surfaces of A2 and B3 respectively. However, the relative errors for the total surfaces of objects A and B were about 0.0036% and 0.07% respectively.

These analyses for volume calculations are also given in Table. 5. The maximum relative error was approximately 0.13% for the geometry of A and about 0.2% for the geometry of B.

4 Conclusions

The radiation flux in Monte Carlo transport problems is often estimated as the track length per unit volume or is related to the current per unit surface area. However, the algorithm for computing cell volumes and surface areas in MCNP and OpenMC is not capable of treating all kinds of geometries, especially for non-rotationally symmetric or non-polyhedral cells.

In this work, two modules named MCSC and MCVC have been proposed and implemented to calculate the surface area and volume of desired cells in the radiation transport codes system. The proposed approach is able to calculate all kinds of surface areas and volumes of complex geometries with reasonable accuracy and precision in a relatively low computational time.

Four simple geometries were investigated as illustrative examples to evaluate the performance of the proposed method. The maximum computation time to obtain a relative error of less than 0.1% in the calculation of volume and surface area of a cubic geometry was about 3.09 sec and 3.52 sec respectively. The results showed that the number of geometrical surfaces can be the most significant factor in determining the elapsed time of calculations.

The presented modules can be efficiently used in all of the Monte Carlo-based fluid dynamics and molecular dy-

namics calculation codes that utilize CG geometry modelling.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work.

References

Breton, E., Choquet, P., Bergua, L., et al. (2008). In vivo peritoneal surface area measurement in rats by microcomputed tomography (μct). *Peritoneal Dialysis International*, 28(2):188–194.

Brown, F. B., Barrett, R., Booth, T., et al. (2002). MCNP version 5. *Trans. Am. Nucl. Soc*, 87(273):02–3935.

Brown, F. B. et al. (2003). MCNP–a general monte carlo n-particle transport code, version 5. *Los Alamos National Laboratory, Oak Ridge, TN*.

Castro, F. and Sbert, M. (2000). Application of Quasi-Monte Carlo Sampling to the Multi Path Method for Radiosity. In *Proceedings of a Conference held at the Claremont Graduate University, Claremont, California, USA*, pages 163–176. Springer.

Do Carmo, M. P. (2016). *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications.

Dorst, L. and Smeulders, A. W. (1987). Length estimators for digitized contours. *Computer Vision, Graphics, and Image Processing*, 40(3):311–333.

Fang, Y., Liu, Y.-S., and Ramani, K. (2009). Three dimensional shape comparison of flexible proteins using the local-diameter descriptor. *BMC Structural Biology*, 9:1–15.

Flin, F., Brzoska, J.-B., Coeurjolly, D., et al. (2005). Adaptive estimation of normals and surface area for discrete 3-D objects: application to snow binary data from X-ray tomography. *IEEE Transactions on Image Processing*, 14(5):585–596.

Gärtner, B. (1999). Fast and robust smallest enclosing balls. In *European Symposium on Algorithms*, pages 325–338. Springer.

Goorley, T., James, M., Booth, T., et al. (2014). Features of MCNP6. In *SNA+ MC 2013-joint international conference on supercomputing in nuclear applications+ Monte Carlo*, page 06011. EDP Sciences.

Hendricks, J. (1980). Calculation of cell volumes and surface areas in MCNP. Technical report, Los Alamos Scientific Lab.

Kargaran, H., Minuchehr, A., and Zolfaghari, A. (2016). The development of GPU-based parallel PRNG for Monte Carlo applications in CUDA Fortran. *AIP Advances*, 6(4).

Klette, R. and Rosenfeld, A. (2004). *Digital geometry: Geometric methods for digital picture analysis*. Morgan Kaufmann.

Klette, R. and Sun, H. J. (2001). Digital planar segment based polyhedrization for surface area estimation. In *International Workshop on Visual Form*, pages 356–366. Springer.

Liu, Y.-S., Yong, J.-H., Zhang, H., et al. (2006). A quasi-Monte Carlo method for computing areas of point-sampled surfaces. *Computer-Aided Design*, 38(1):55–68.

Lorensen, W. E. and Cline, H. E. (1998). Marching cubes: A high resolution 3D surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353.

McCloskey, S. E. and Braithwaite, W. J. (1995). Introduction to Monte Carlo methods. *Journal of the Arkansas Academy of Science*, 49(1):109–114.

Prisant, M. (1996). Applications of the ray representation to problems of protein structure and function. In *Proceedings of CSG*, pages 33–47.

Romano, P. K. and Forget, B. (2013). The OpenMC Monte Carlo particle transport code. *Annals of Nuclear Energy*, 51:274–281.

Romano, P. K., Horelik, N. E., Herman, B. R., et al. (2015). OpenMC: A state-of-the-art Monte Carlo code for research and development. *Annals of Nuclear Energy*, 82:90–97.

Roth, S. D. (1982). Ray casting for modeling solids. *Computer Graphics and Image Processing*, 18(2):109–144.

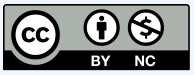
Sarraga, R. F. (1982). Computation of surface areas in GM-Solid. *IEEE Computer Graphics and Applications*, 2(07):65–70.

Schaufler, G. and Jensen, H. W. (2000). Ray tracing point sampled geometry. In *Rendering Techniques 2000: Proceedings of the Eurographics Workshop in Brno, Czech Republic, June 26–28, 2000 11*, pages 319–328. Springer.

Timmer, H. and Stern, J. (1980). Computation of global geometric properties of solid objects. *Computer-Aided Design*, 12(6):301–304.

Windreich, G., Kiryati, N., and Lohmann, G. (2003). Voxel-based surface area estimation: from theory to practice. *Pattern Recognition*, 36(11):2531–2541.

Zeng, X., Staib, L. H., Schultz, R. T., et al. (1999). Segmentation and measurement of the cortex from 3-D MR images using coupled-surfaces propagation. *IEEE Transactions on Medical Imaging*, 18(10):927–937.



To cite this article:

Kargaran, H. (2024). A comprehensive approach for calculation of surface area and volume in radiation transport codes.*Radiation Physics and Engineering*, 5(2), 61-69.

DOI: 10.22034/rpe.2024.418185.1164

To link to this article: https://doi.org/10.22034/rpe.2024.418185.1164