

A Deep Penetration Problem Calculation Using AETIUS: An Easy Modeling Discrete Ordinates Transport Code UsIng Unstructured Tetrahedral Mesh, Shared Memory Parallel

Jong Woon KIM ^{1*} and Young-Ouk LEE ¹

¹Korea Atomic Energy Research Institute, 989 Daeduck-daero, Yuseong-gu, Daejeon 305-353, Korea

*Corresponding Author, E-mail: jwkim@kaeri.re.kr

Abstract

As computing power gets better and better, computer codes that use a deterministic method seem to be less useful than those using the Monte Carlo method. In addition, users do not like to think about space, angles, and energy discretization for deterministic codes.

However, a deterministic method is still powerful in that we can obtain a solution of the flux throughout the problem, particularly as when particles can barely penetrate, such as in a deep penetration problem with small detection volumes.

Recently, a new state-of-the-art discrete-ordinates code, ATTILA, was developed and has been widely used in several applications. ATTILA provides the capabilities to solve geometrically complex 3-D transport problems by using an unstructured tetrahedral mesh.

Since 2009, we have been developing our own code by benchmarking ATTILA. AETIUS is a discrete ordinates code that uses an unstructured tetrahedral mesh such as ATTILA. For pre- and post- processing, Gmsh is used to generate an unstructured tetrahedral mesh by importing a CAD file (*.step) and visualizing the calculation results of AETIUS. Using a CAD tool, the geometry can be modeled very easily.

In this paper, we describe a brief overview of AETIUS and provide numerical results from both AETIUS and a Monte Carlo code, MCNP5, in a deep penetration problem with small detection volumes. The results demonstrate the effectiveness and efficiency of AETIUS for such calculations.

KEYWORDS: deterministic code, discrete ordinates method, unstructured tetrahedral mesh, deep penetration, high S_N order, comparison

I. Introduction

These days, when we develop or use a deterministic code, a lot of people ask what is its advantage compared to computer codes that use a Monte Carlo method such as MCNP5⁽¹⁾.

The reason for its use is a continual increase in computing power and parallel processing capabilities. Users prefer to use a method that has less approximation such as a Monte Carlo method.

Even for the deterministic codes, the results of a Monte Carlo codes are used as a reference calculation for cross checking.

Then, does a deterministic method have any better features than a Monte Carlo method? This is the main motivation of this paper and why we chose a deep penetration problem with small detection volumes.

In this paper, we describe a brief overview of AETIUS and provide numerical results from both AETIUS and a Monte Carlo code, MCNP5, in a deep penetration problem with small detection volumes.

II. Method

Before naming our code as AETIUS (An Easy modeling Transport code usIng Unstructured tetrahedral mesh, Shared memory parallel), it was tested using several applications⁽²⁻⁴⁾. MUST (Multi-group Unstructured geometry S_N Transport) is a twin code that uses C++^(5,6).

AETIUS is programmed using f90 and uses Gmsh⁽⁷⁾ as a pre- and post- processing program. A white boundary condition on an arbitrary surface, the first collision source on the volume source, and shared memory parallel capabilities have been added to AETIUS.

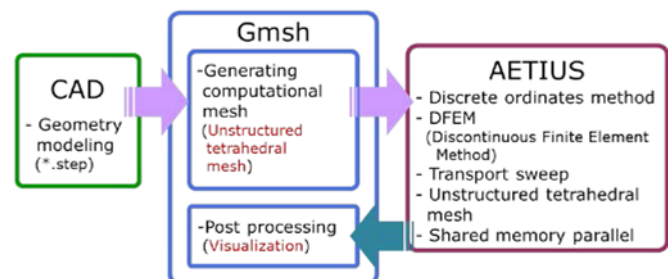


Figure 1: The overall calculation flow of AETIUS

Compare to the deterministic codes use a regular mesh, one of the merits of MCNP5 is its capability to deal with a complicated geometry without any computational burden.

However, with the help of a CAD tools (e.g., CATIA) and Gmsh, we can finish complicated geometry modeling for AETIUS very easily even though the geometry is a helical tube in a cube as shown in Figure 2. This modeling is difficult to accomplish without using a CAD tool and unstructured tetrahedral mesh.

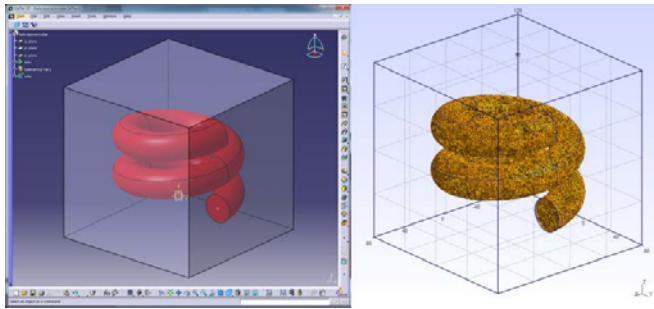


Figure 2: An example of easy modeling: (left) a CAD modeling with CATIA and (right) an unstructured tetrahedral mesh with Gmsh

1. Space

Every deterministic code requires spatial discretization to solve the Boltzmann transport equation. An unstructured tetrahedral mesh has been used for structural mechanics, fluid dynamics, and so on. However, it has not been widely used for nuclear reactor designs or radiation shielding problems.

To deal with an unstructured tetrahedral mesh in the discrete ordinates transport code, Wareing T.A. et al. proposed a discontinuous finite element method (DFEM)⁽⁸⁾ in 2001 and developed the ATTILA⁽⁹⁾ code.

AETIUS also uses DFEM for space discretization. DFEM is used for solving the balance equation between adjacent tetrahedral elements according to the direction of angle in a discrete set.

2. Angle

We recently implemented a triangular and rectangular Chebyshev-Legendre⁽¹⁰⁾ quadrature to AETIUS so that we can increase the angular quadrature up to S₁₀₀.

If the problem is ray-effect dominant, we may obtain better results using the first collision source method. However, if the problem consists of a ray-effect dominant zone (e.g., air) and a scattering dominant zone (e.g., concrete) together, the capability to increase S_N order over S₂₀ may be useful because the first collision source method alone may not be enough to deal with this kind of problem.

3. Energy

We also use a multi-group library. The libraries currently in use⁽¹¹⁾ were generated based on ENDF/B-VII.0 and listed in Table 1.

| Name | Group structure | |
|------|-----------------|-------|
| | Neutron | Gamma |

| | | |
|------------|-----|----|
| VITAMIN-B6 | 199 | 42 |
| BUGLE-96 | 47 | 20 |
| VITAMIN-J | 175 | 42 |
| LANL-30 | 30 | 12 |
| LANL-80 | 80 | 24 |
| SCALE-44 | 44 | 18 |

Table 1: A list of available multi-group libraries for AETIUS

4. Shared memory parallel

AETIUS is a three-dimensional discrete ordinates code, which solves the Boltzmann transport equation on an unstructured tetrahedral mesh with DFEM. DFEM requires an unknown angular flux at four vertices per tetrahedral element. The angle and energy are discretized through the angular quadrature and multi-group structure, respectively.

For example, in the source code, the angular flux is given as Eq. (1), where 4 indicates four vertices per element, elem is the index of tetrahedral element, n is the index of discretized angle, and g is the index of energy group.

$$\psi(4, \text{elem}, n, g) \tag{1}$$

If we have to solve a large 3D problem so that the number of elements, angular quadrature, or energy group need to be increased, then it requires more memory and a longer calculation time according to a variable size increase.

OpenMP⁽¹²⁾ is implemented in AETIUS, and is easy to be implemented in an existing serial code. Besides, AETIUS requires a large amount of memory when solving a large problem, shared memory parallel might be more suitable.

Currently, we run AETIUS on a server that has four Intel Xeon E7-4890v2 2.8Ghz CPUs using 120 threads with 512 Gb of memory.

III. Application

1. Description

To see the effectiveness and efficiency of AETIUS, we chose a deep penetration problem with small detection volumes, as shown in Figure 3.

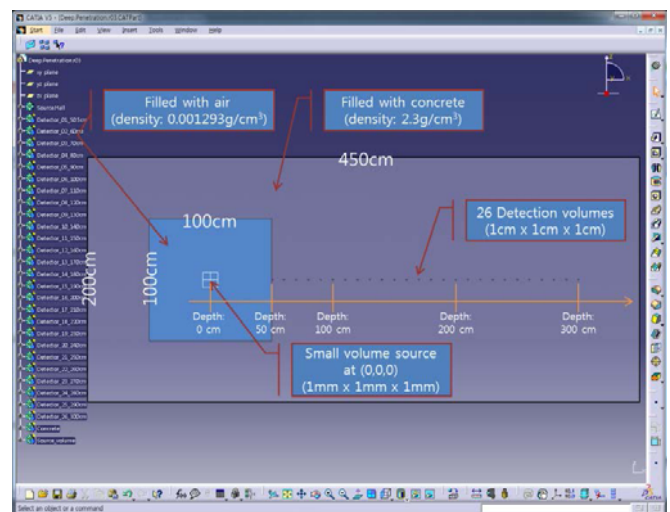


Figure 3: Overview of the deep penetration problem with small detection volumes

A 1m×1m×1m cube is located at the center of origin (0,0,0) and filled with air. Concrete covers the outside of the cube, as shown in Figure 3. The thickness in the x-direction is 3m and in the other directions is 50cm.

A small volume source zone (1mm³) is prepared for one of two AETIUS calculations. 26 detection volumes (1cm×1cm×1cm) are located along the x-axis from 50cm to 300cm with 10cm spacing.

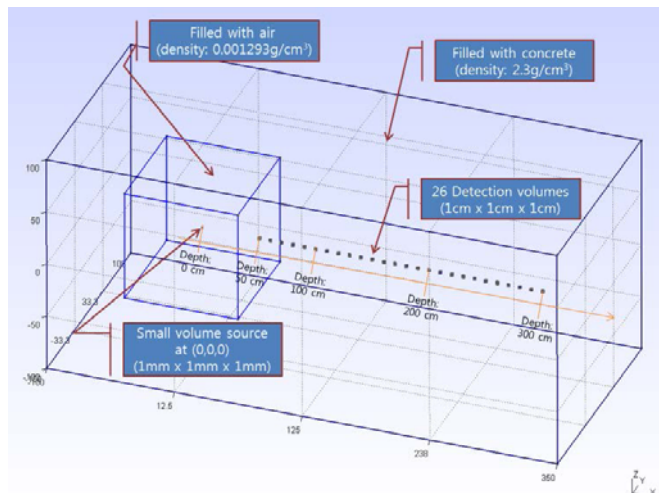


Figure 4: Three-dimensional view of a deep penetration problem with small detection volumes

The computational mesh is generated by Gmsh and shown in Figure 5.

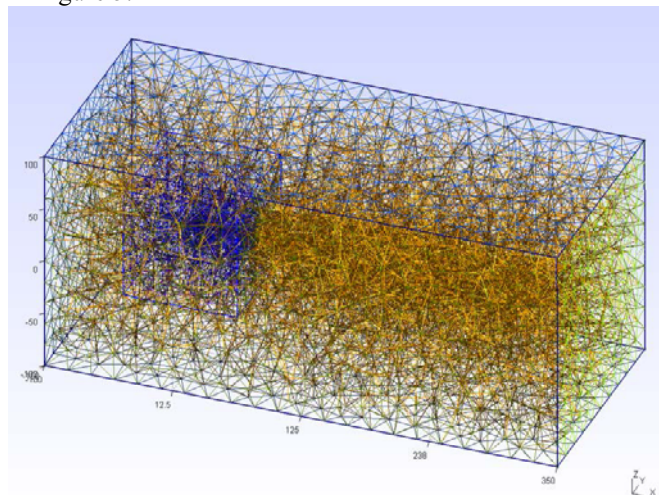


Figure 5: The computational mesh (generated by Gmsh, 61,562 unstructured tetrahedrons)

Transport calculations are performed on a deep penetration problem with small detection volumes. This problem seems very simple and easy. However, in view of the Monte Carlo transport code, it may be simple but not easy to obtain solutions with a reliable confidence interval (the relative error is less than 0.1 for all other tallies)⁽¹⁾.

It is hard to send particles deep into the concrete block and the tally zones (detection volumes) are small (1cm³).

To obtain a satisfactory solution with MCNP5, variance reduction techniques such as splitting or weight window

should be used.

In MCNP5 calculation, the neutron source is evenly sampled between 15MeV and 17MeV, which is the 1st group of LANL-30 group structure. Some other calculation parameters are listed in Table 2.

| | MCNP5 | AETIUS | |
|------------------------|--|---|---|
| | | with FCS | without FCS |
| Source strength | Point source: 1 source particle/sec at origin (0,0,0) | Point source: 1 source particle/sec at origin (0,0,0) | Volume source: 10 ³ source particle/cm ³ -sec, volume source zone (10 ⁻³ cm ³) |
| Source spectrum | Source is given in the 1 st group of LANL-30 | | |
| Library | ENDF-B/VII.0 | | |
| Energy group structure | Continuous energy | LANL-30 | |
| Material density | Air : 0.001293 g/cm ³ Concrete : 2.3 g/cm ³ | | |
| P _N order | n/a | 3 | |
| S _N order | n/a | Triangular Chebyshev-Legendre S ₄₀ | |
| Calc. options | weight window | FCS with point source | Volume source only |
| Error criterion | nps :5×10 ⁹ | 1×10 ⁻⁴ | |
| Parallel | MPI (50 cores) | OpenMP (120 cores) | |

FCS : first collision source method

Table 2: Calculation parameters

2. Results

The calculated total neutron flux from AETIUS is visualized through Gmsh and shown in Figure 6. This is a sectional view of the lower part of the total neutron flux isosurface distribution.

One of the key advantages of using a discrete ordinates code such as AETIUS is that we can obtain a solution of the flux throughout the problem. This is particularly useful when we wish to understand how the flux changes throughout the modelled geometry.

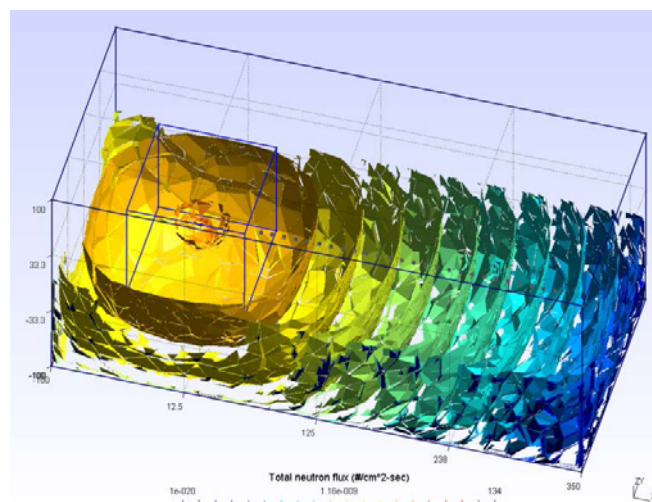


Figure 6: Visualized total neutron flux isosurface distribution (sectional view of the lower part)

Volume averaged fluxes on the 26 detector volumes are compared with those of MCNP5 and are shown in Figure 7. These fluxes of AETIUS were obtained through a single run. However, for MCNP5, we have to run it several times to obtain the weight window and re-run it again. Although we used weight window in the MCNP5 calculation, the average fluxes in the last two detection volumes could not be obtained.

In view of relative error, volume-averaged fluxes for the 17 detection volumes near source area (from 50cm to 210cm) are within a reliable confidence interval and are well matched with those of AETIUS.

However, for last 9 detection volumes, the relative errors of MCNP5 result are getting increased as depth is increased. Moreover, relative errors of last two detection volumes are zero since no particles are tallied.

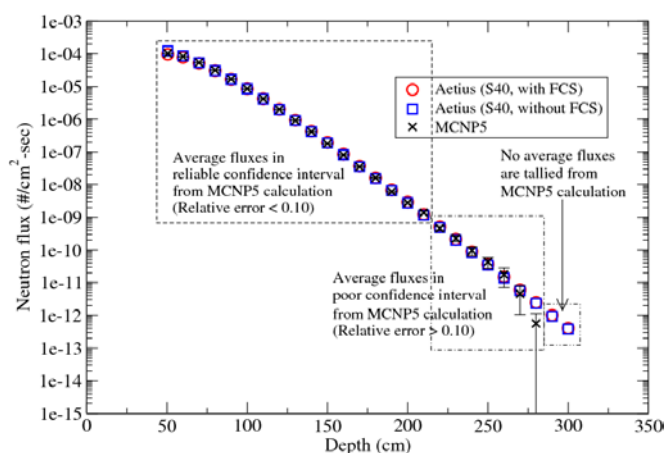


Figure 7: Comparison of volume averaged fluxes on 26 detection volumes

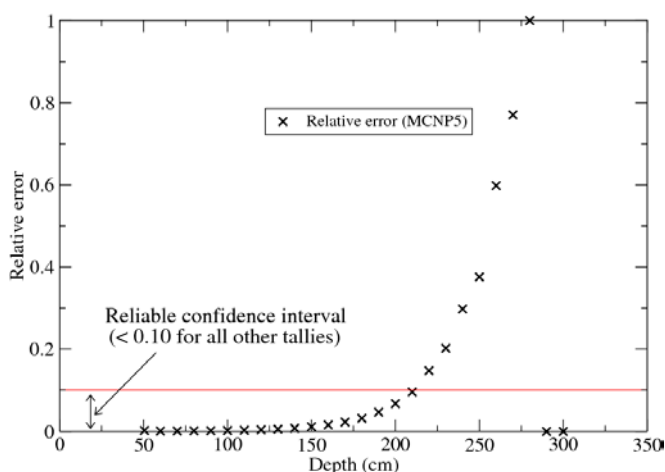


Figure 8: Relative error of MCNP5 on 26 detection volumes

IV. Conclusion

We described a brief overview of AETIUS and provided numerical results from both AETIUS and MCNP5 for a deep penetration problem with small detection volumes.

Although the results of the two codes are obtained using completely different calculation methods, they show a good agreement.

Deterministic codes (e.g., AETIUS) require many spatial elements, high S_N/P_N orders, and many energy groups to obtain a good solution, which could require many iterations and longer calculation time.

Similarly, Monte Carlo codes (e.g., MCNP5) also require longer running time and many iterations to optimize the variance reduction techniques (e.g., splitting and weight window).

However, deterministic codes can provide a rough estimate of the flux throughout a problem relatively quickly. This can help to identify areas of a problem where shielding may or may not be performing as well as expected.

References

- 1) X-5 Monte Carlo Team, *MCNP - A General Monte Carlo N-Particle Transport Code, Version 5, Volume II: User's Guide*, LA-CP-03-0245, Los Alamos National Laboratory (2008).
- 2) J.W. Kim, S.G. Hong, and Y. Lee, "Computational efficiency of a modified scattering kernel for full-coupled photon-electron transport parallel computing with unstructured tetrahedral meshes," *Nuclear Engineering and Technology*, vol. 46, no. 2, 263-272 (2014).
- 3) J.W. Kim, C.W. Lee, Y. Lee, D. Lee, and S. Cho, "Development of discrete ordinates code supporting unstructured tetrahedral mesh and applied in neutronics analysis for the Korea Helium Cooled Ceramic Reflector Test Blanket Module," *Fusion Engineering and Design*, 89, 1172-1176 (2014).
- 4) J.W. Kim, C.W. Lee, Y. Lee, D. Lee, and S. Cho, "Preliminary study on applying discrete ordinates code supporting unstructured tetrahedral mesh to the 40-degree toroidal segment ITER model," *Fusion Science and Technology*, 68, 652-656 (2015).
- 5) S.G. Hong, J.W. Kim, and Y. Lee, "Development of MUST (Multi-group Unstructured geometry S_N Transport) Code," *Transaction of the Korean Nuclear Society Autumn Meeting*, Gyeongju, Korea (2009).
- 6) J.W. Kim, S.G. Hong, and Y. Lee, "MUST code verification and validation on the shielding test problems," *ICRS-12 & RPSD-2012*, Nara, Japan (2012).
- 7) C. Geuzaine, J.-F. Remacle, "Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities," *Int. J. Numer. Meth. Eng.* 79 (11) 1309-1331 (2009).
- 8) T.A. Wareing, J.M. McGhee, J.E. Morel, and S.D. Pautz, "Discontinuous finite element S_N methods on three-dimensional unstructured grids," *Nuclear Science and Engineering*, 138, 256-268 (2001).
- 9) J.M. McGhee, T.A. Wareing, and D.A. Barnett Jr., *Attila User's Manual*, Transpire Inc. (2007).
- 10) K.D. Lathrop and B.G. Carlson, *Discrete ordinates angular quadrature of the neutron transport equation*, LA-3186, Los Alamos National Laboratory (1965).
- 11) D.H. Kim, C. Gil, and Y. Lee, "Validation of an ENDF/B-VII.0-based neutron and photon shielding library in MATXS-format," *Journal of the Korean Physical Society*, vol. 59, no. 2, 1199-1202 (2011).
- 12) Barbara Chapman et al., *Using OpenMP*, The MIT Press, Cambridge, Massachusetts, London, England (2008).