M&C 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering, Jeju, Korea, April 16-20, 2017, on USB (2017)

SERPENT Performance with Hybrid Combinatorial and Stereolithographic Geometry*

A. Talamo[1a], Y. Gohar[a], and J. Leppänen[b]

[a)] *Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439, USA*
[b)] *VTT Technical Research Centre, P.O. Box 1000, FI-02044 VTT, Finland*
[1)] *alby@anl.gov*

Abstract – *The GIACINT critical facility of Belarus has been modeled by the SERPENT and MCNP Monte Carlo codes. The latter code uses combinatorial geometry whereas the first code uses either stereolithography geometry or hybrid geometry. In the hybrid geometry, the complicated parts of the facility are modeled by stereolithography geometry and the simple parts by combinatorial geometry.*

I. INTRODUCTION

This study analyzes the experiments performed at the GIACINT facility of Belarus [1,2] with the Monte Carlo code SERPENT [3,4]. The critical configuration of this facility has been modeled using both combinatorial and stereolithography (STL) [5] geometry. With combinatorial geometry, the facility parts are modeled using Boolean operations (e.g. union, intersection, and complement) on basic volumes (e.g. cylinders, prisms, hexagons, etc.). In addition, the facility parts can be arranged on a regular lattice to model arrays of fuel rods or assemblies. With stereolithography geometry, the facility parts are modeled using triangular surfaces (facets) without the possibility to define any lattice.

Both combinatorial and stereolithography geometries allow using universes. A universe is a piece of geometry that has its own coordinates system and that can be nested into the real geometry or into another universe.

Six different SERPENT models of the GIACINT facility have been developed with the purpose of reducing the computing time. These models use either a stereolithography geometry or a hybrid geometry mixing combinatorial and stereolithography geometries together. Obviously, a computational model based on combinatorial geometry runs faster than one based on the stereolithpography geometry. Consequently, using the hybrid model can significantly reduce the computing time relative to a pure stereolitography geometry model. In an optimized hybrid model, the complicated parts of the facility are modeled using the stereolithography geometry, whereas the simple parts are modeled using combinatorial geometry and lattices.

The results obtained by SERPENT have been compared with those a MCNP [6] computation based on combinatorial geometry.

II. THE GIACINT FACILITY

The fuel material used in the GIACINT facility is uranium-zirconium carbonitride ($U_{0.9}Zr_{0.1}C_{0.5}N_{0.5}$) with 11.9 g/cm^3 density and 19.75% uranium enrichment. The active radius and height of the fuel rods are 5.375 mm and 50 cm, respectively. The total fuel rod length is 62 cm and the thickness of the helium gap and stainless steel (type 06X18H10T) clad are 25 μm and 0.6 mm, respectively. The fuel rods are arranged into a hexagonal matrix, with 1.8, 3.2, or 4.7 cm fuel pitch, and are supported by a solid stainless steel (type 12X18H10T) hexagonal plate. The latter is connected to the tank through six stainless steel support rods. In addition to the bottom support plate, two aluminum hexagonal grids and a top stainless steel hexagonal plate, which has stainless steel plugs filling the empty fuel holes, hold the fuel rods together. The top stainless steel support plate and two aluminum grids have rounded-rectangular holes to house twelve detector channels. The latter penetrate the assembly from the top and reach the upper surface of the bottom support plate. Each detector hole has a cylindrical aluminum guide covered by plexiglas. The steel tank has a cylindrical hole starting from the tank bottom surface up to the bottom surface of the bottom support plate to allow the insertion of the californium neutron source igniting the fission chain. Figure 1 illustrates a picture of the assembly.

The facility is equipped with six pairs of absorbing elements, which penetrate the active core, from the top, only during an emergency condition. The absorbing elements have the same dimensions as the fuel rods and contain boron carbide instead of uranium fuel. Three pairs of absorber elements are connected to the top surface of fuel rods through a stainless steel cylindrical connector with 8 cm height and 3.1 mm radius. The other three pairs of absorber rods are connected to plexiglas rods, which are inserted into the active zone of the core during normal operations. Under an accident scenario, the six pairs of

The submitted manuscript has been created by UChicago Argonne LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357.

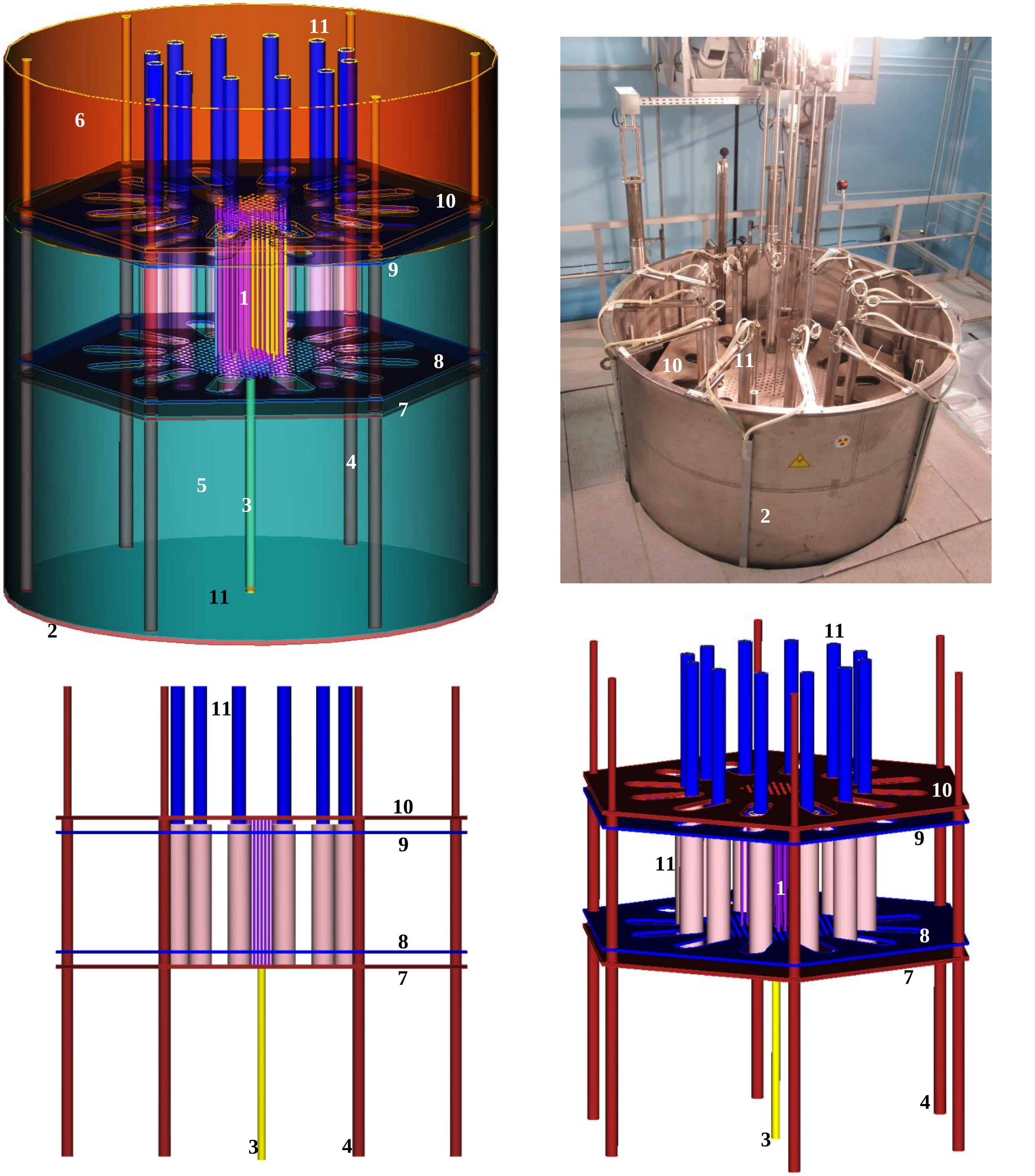

Fig. 1. GIACINT critical assembly: overview of the CUBIT model (left-top plot), overview of the real facility (rigth-top plot), frontal view of the CUBIT model without the steel tank and water (left-bottom plot), and overview of the CUBIT model without the steel tank and water (right-bottom plot). Legend: 1) fuel rods (uranium fuel, stainless steel clad, and helium gap); 2) stainless steel tank; 3) channel for californium source (air); 4) Stainless steel support rods; 5) water moderator; 6) assembly top zone (air); 7) stainless steel bottom support plate; 8) aluminum fuel rods bottom grid; 9) aluminum fuel rods top grid; 10) stainless steel top support plate; 11) detectors channels (air, aluminum, and plexiglas).

fuel and plexiglas rods shift axially down and let the absorber rods penetrate the active zone of the core. The critical assembly tank is disposed into a room with 5.9 x 7.4 m area and 8.5 m height; the concrete walls of the room have thickness ranging from 20 cm up to 1.5 m.

The GIACINT facility can be operated using a different number of fuel rods and pitches; the latter are changed by replacing the stainless steel plates and aluminum grids. The criticality condition is set by varying the water level in the assembly tank. The water level is measured starting from the bottom surface of the active fuel zone (from the top surface of the bottom stainless steel plate). In this study, the facility has been loaded with 66 fuel rods.

III. SERPENT STL GEOMETRY MODELING

SERPENT is a general purpose neutron and photon transport code developed since 2004 at VTT Technical Research Centre (Finland) and written in C language. The source code consists of 195,000 lines. This Monte Carlo code offer the possibility to model a reactor core either by using the traditional combinatorial geometry, widely used in the MCNP code, or by using the stereolithography geometry. This latter feature is unique of the SERPENT code.

In the STL geometry representation, volumes are defined by the union of triangular surfaces (facets). The typical content of a STL file is:

```
solid Body_158
      facet normal -9.999247e-001 1.227134e-002 0.000000e+000
      outer loop
         vertex 1.062500e+000 1.385641e+001 2.500000e+001
         vertex 1.062662e+000 1.386960e+001 2.500000e+001
         vertex 1.062662e+000 1.386960e+001 -2.500000e+001
      endloop
      endfacet
      facet normal -9.999247e-001 1.227134e-002 0.000000e+000
      outer loop
         vertex 1.062500e+000 1.385641e+001 -2.500000e+001
         vertex 1.062500e+000 1.385641e+001 2.500000e+001
         vertex 1.062662e+000 1.386960e+001 -2.500000e+001
      endloop
      endfacet
      facet normal -9.999247e-001 -1.227134e-002 0.000000e+000
      outer loop
         vertex 1.062500e+000 1.385641e+001 -2.500000e+001
         vertex 1.062662e+000 1.384322e+001 2.500000e+001
         vertex 1.062500e+000 1.385641e+001 2.500000e+001
      endloop
      endfacet
      ....
      ....
      ....
      ....
endsolid Body_158
solid Body_159
      ....
      ....
      ....
      ....
endsolid Body_159
```

The above example defines bodies (volumes) #158 and #159 and lists the vertexes of the triangular surfaces for body #158 and their associated normal vectors. In the STL geometry model, the whole geometry is defined by the sequence of *solid* and *endsolid* sections. In the STL file, each volume of the geometry must have a *solid* and *endsolid* section. The precision of the STL file, relative to the real geometry, depends on the total number of triangular surfaces. Clearly, more triangular surfaces provide better representation of the real geometry, especially when the volumes have non-planar boundaries.

There are many benefits in using the STL geometry, including:

1) the CAD geometry model of the core or any part of the core can be directly imported in SERPENT;
2) the neutronics calculation can be easily coupled to the thermal-hydraulics calculation using a single SERPENT/OPENFOAM executable without any external interface;
3) the model can be printed by 3D printers.

The STL geometry can be constructed using the CUBIT software [7]. The latter is a solid modeling and finite elements mesh generator software. Since version 14.1, CUBIT allows defining the *feature angle* parameter before writing the STL file. This angle is formed by two tangents starting from two consecutive vertexes on a curve of the real geometry. A lower *feature angle* increases the precision of the STL geometry, and the total number of triangular surfaces, relative to the real geometry, at the expense of the STL file size.

In the STL geometry modeling, the materials to volumes mapping can be performed by PYTHON scripts and a C code, as described in Ref. 1. At present, this mapping is not performed by SERPENT.

In this study, the results and the computing time of six different SERPENT geometry models have been compared with those from MCNP version 6.1.1β [6]. The MCNP code uses only combinatorial geometry. Both SERPENT and MCNP modeled the GIACINT experimental facility of Belarus, as described in Refs. 1 and 2. The computational analyses used the ENDF/B-VII.0 nuclear data library [8].

When SERPENT uses the STL geometry, it defines a regular lattice of cells superimposed to the STL geometry and starts a ray-tracing procedure from the triangular surfaces. This ray-tracing procedure defines the volume that a neutron enters after crossing a triangular facet (mesh search). The adaptive search mesh algorithm of SERPENT attaches to each STL facet a list of the intersected regular lattice cells. The latter define a rectangular bounding box containing the minimum and maximum values of the three vertices of the triangular facet. When a triangular facet is large (e.g. the *feature angle* is equal to 4 or 5), it has attached a long list of regular lattice cells.

The six STL geometry configurations modeled by SERPENT include:

1) reference modeling;
2) two STL universes modeling;
3) hybrid modeling;
4) hybrid modeling with lattice;
5) hybrid modeling with lattice and optimized search mesh;
6) hybrid modeling with lattice, optimized search mesh, and background universe with fuel.

In all the six STL files, the CUBIT *feature angle* parameter has been set to 2. In the reference STL model, the whole geometry was modeled by a single STL universe embedded in a background universe made of air. The reference model has two universes. A universe is a piece of geometry that can fill and be filled by another piece of geometry. The *universe* concept in Monte Carlo neutronics codes corresponds to the *body* concept in computer aided design (CAD) modeling for mechanical and thermal-hydraulics calculations. The STL universe has been defined by the SERPENT card:

```
solid  2 1 bg  % type, universe, background universe
10 4 5 4 3 2  % split criterion, split levels, splits per level
```

The first three parameters of the *solid* card represent: the type (STL geometry requires a value equal to 2), universe number, and background universe name/number (bg) of the STL piece of geometry. Unlike MCNP, SERPENT accepts alphanumeric characters for the universes label, whereas MCNP allows only integer numbers. The second line below the *solid* card contains: the split criterion (10), the number of split levels in the adaptive search mesh (4), and the number of splits at each level (5 4 3 2). The number of entries following the split levels value must be equal to the split levels value. The split criterion represents the number of facets in the adaptive search mesh that causes the cell to be split. A value of 4 for the number of split levels means that each cell of the adaptive search mesh can be split up to 4 times. In the above SERPENT cards, first the whole geometry is split into 5 cells; if these cells contain triangular surfaces, they are split 4 times; if the resulting cells contain triangular surfaces, they are split 3 times; if the resulting cells contain triangular surfaces, they are split 2 times.

In the second model, two different STL universes are used. One STL universe is used for the fuel zone and is embedded in a water and air non-STL universe. The other STL universe is used for the other parts of the geometry and is embedded into the air background universe. The 6 different SERPENT models used the same triangular surfaces, as generated by CUBIT. The SERPENT cards for model 2 are:

```
solid  2 1 2      % type, universe, background universe
1 5 20 2 2 2 2  % split criterion, split levels, splits per level
solid  2 2 bg   % type, universe, background universe
1 6 2 2 2 2 2 2 % split criterion, split levels, splits per level
```

In the third SERPENT model the fuel rods have been defined, one by one, with combinatorial geometry. This SERPENT model uses the card:

```
solid  2 1 bg     % type, universe, background universe
1 6 2 2 2 2 2 2  % split criterion, split levels, splits per level
```

In the fourth SERPENT model, the fuel rods have been defined with combinatorial geometry using a hexagonal lattice. This SERPENT model uses the same *solid* card as model 3. Defining the geometry by a lattice is a feature of Monte Carlo neutronics codes; deterministic neutronics, mechanics, and thermal-hydraulics codes do not have this capability.

The fifth SERPENT model is equal to the fourth model with the exception of using optimized parameters for the adaptive search mesh, as defined in the following *solid* card.

```
solid  2 1 bg    % type, universe, background universe
1 4 10 5 4 3     % split criterion, split levels, splits per level
```

The sixth SERPENT model is the same as the fifth model with the exception that the lattice of fuel rods has been defined in the background non-STL (combinatorial geometry) universe rather than in a separate non-STL universe.

All Monte Carlo computations have been performed on a Linux node with 32 cores with no Ethernet exchange of data. The node runs on Linux CentOS version 7 operating system and has 125 Gb RAM memory. Each core of the node consists of an Intel Xeon processor with 2.27 GHz frequency and 24,576 kb cache memory. MCNP has been compiled by Intel compiler version 15.0.4 and SERPENT has been compiled by gcc version 4.8.5. MCNP parallel communications have been performed by MPICH version 3.2.0, whereas SERPENT parallel communications have been performed by OPENMP (as embedded in the gcc compiler). Both MCNP and SERPENT computations used 31 cores for neutron transport simulations. This number of cores has been selected because the master core in MPI computations of MCNP does not simulate neutron transport, whereas in OPENMP computations there is no master core.

The SERPENT reference input consists of 696 lines, including 471 lines written by a 160 lines C program [3]. The CUBIT input file has 458 bodies (volumes), 4448 surfaces, 6156 curves, and 5820 vertices. The CUBIT input consists of 3980 lines, including 3638 lines written by a 105 lines MATLAB script. A PYTHON script (6 lines) has been used to relate colors with volumes in the CUBIT model and a C program (160 lines) has been used to map the colors of the CUBIT model into different materials in SERPENT [3]. Consequently, the SERPENT modeling of the GIACINT facility required 838 handwritten input lines distributed over CUBIT, MATLAB, PYTHON, and C computational tools.

IV. RESULTS

The results and geometry plots of the SERPENT simulations are reported in Table 1 and illustrated in Figs. 2 to 13. Splitting the STL geometry into two zones, one for the fuel and the other for the rest of the core, reduces SERPENT computational time from 1848 down to 596 minutes. Modeling the fuel zone using a lattice, rather than modeling every single fuel rod, has no significant impact on the computing time (models 3 and 4). Optimizing the adaptive search parameters reduces the computing time from 570 to 180 minutes. Modeling the fuel zone in the background universe, rather than in a separate universe, has a small change in the computing time (models 5 and 6). The optimized SERPENT model and MCNP have similar computing time, 168 and 128 minutes, respectively. SERPENT and MCNP produce similar effective multiplication factors, within 50 pcm.

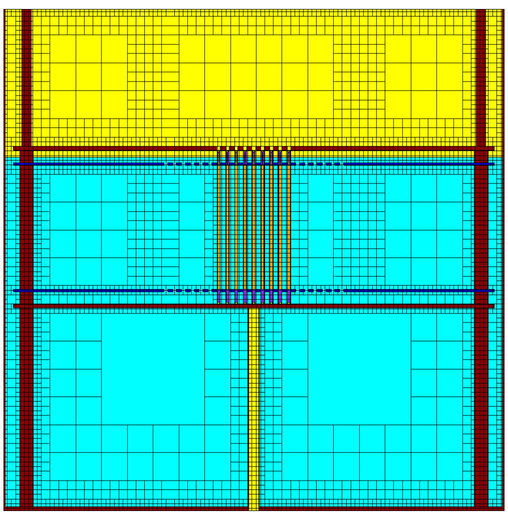

Fig. 2. Vertical section of SERPENT model 1.

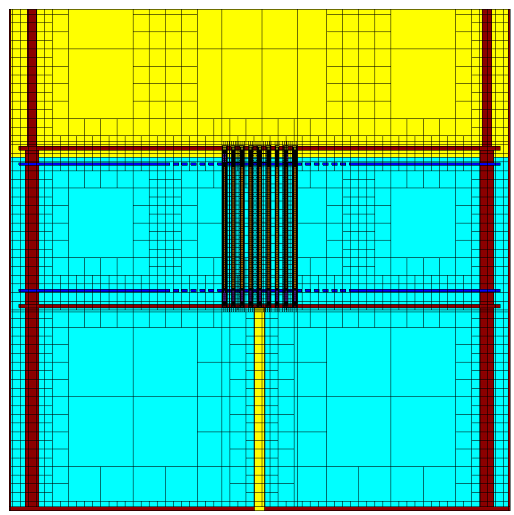

Fig. 4. Vertical section of SERPENT model 2.

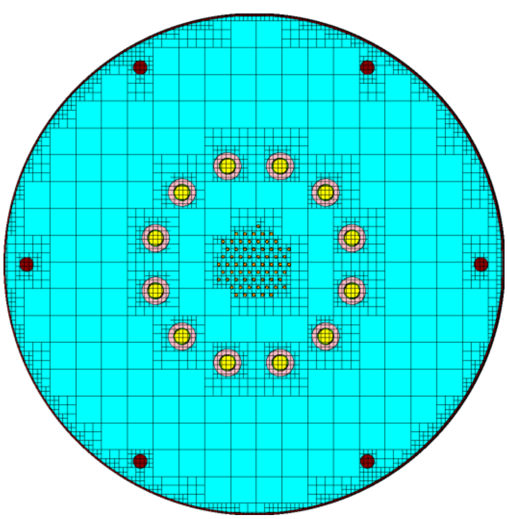

Fig. 3. Horizontal section of SERPENT model 1.

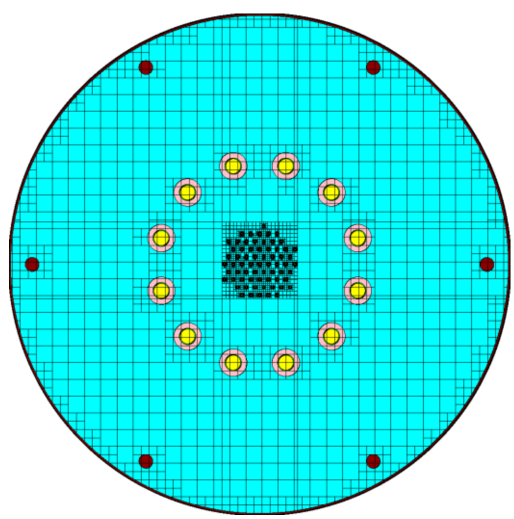

Fig. 5. Horizontal section of SERPENT model 2.

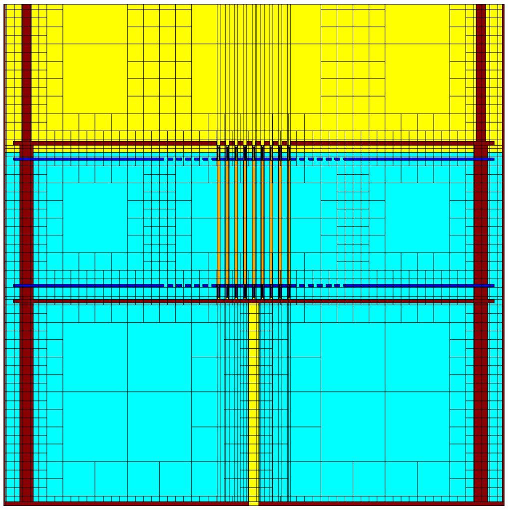

Fig. 6. Vertical section of SERPENT model 3.

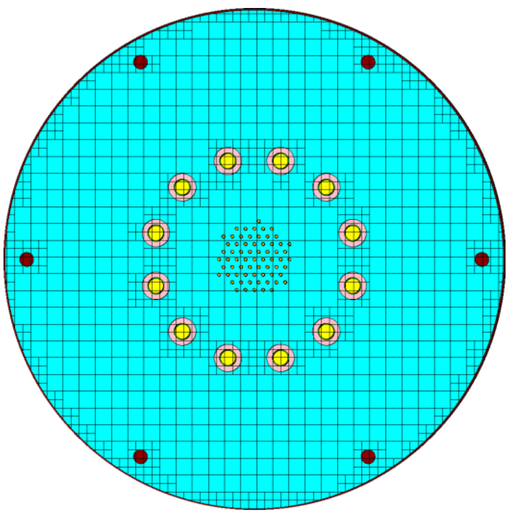

Fig. 7. Horizontal section of SERPENT model 3.

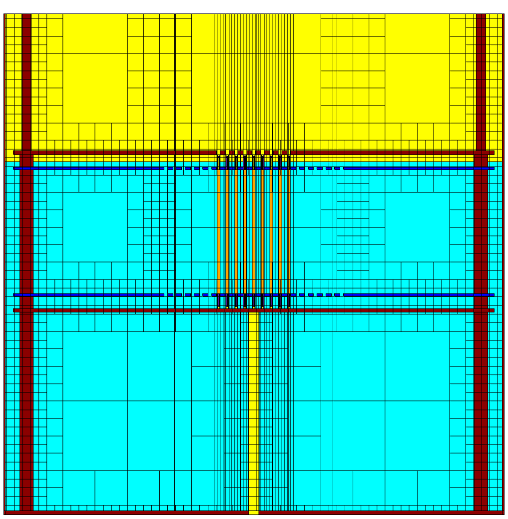

Fig. 8. Vertical section of SERPENT model 4.

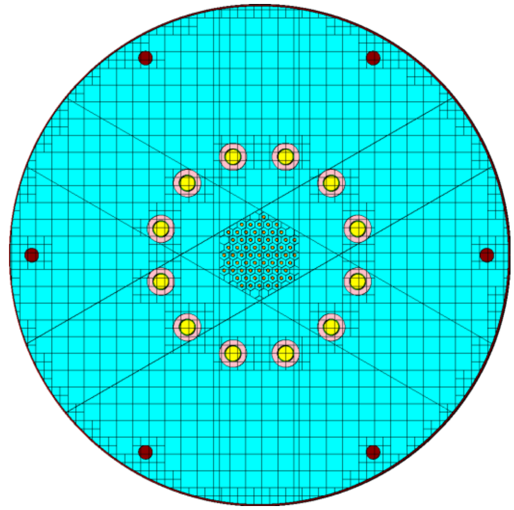

Fig. 9. Horizontal section of SERPENT model 4.

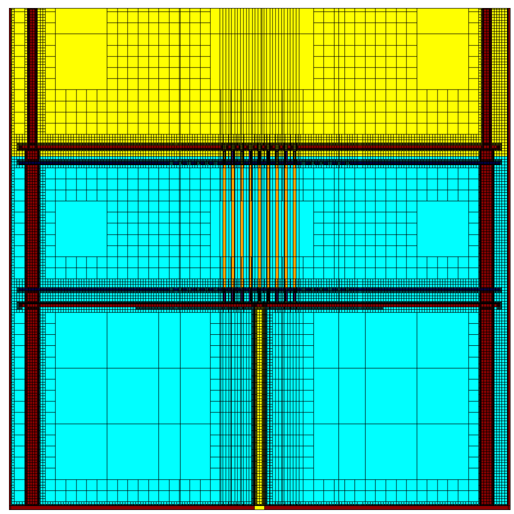

Fig. 10. Vertical section of SERPENT model 5.

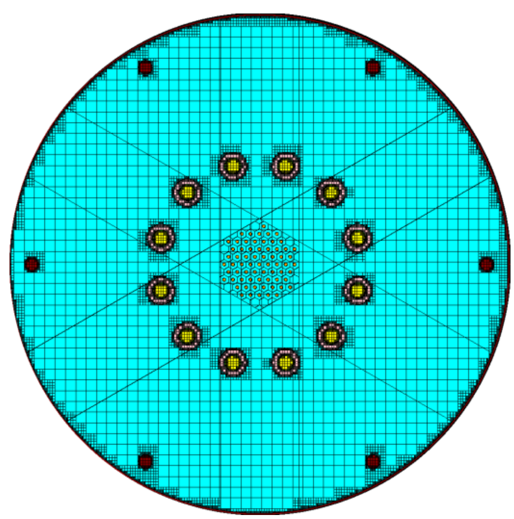

Fig. 11. Horizontal section of SERPENT model 5.

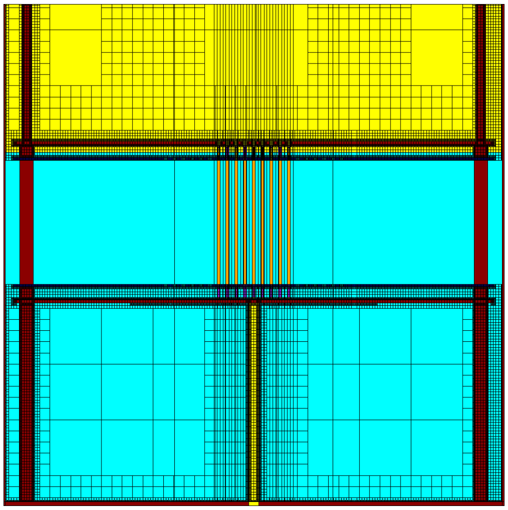

Fig. 12. Vertical section of SERPENT model 6.

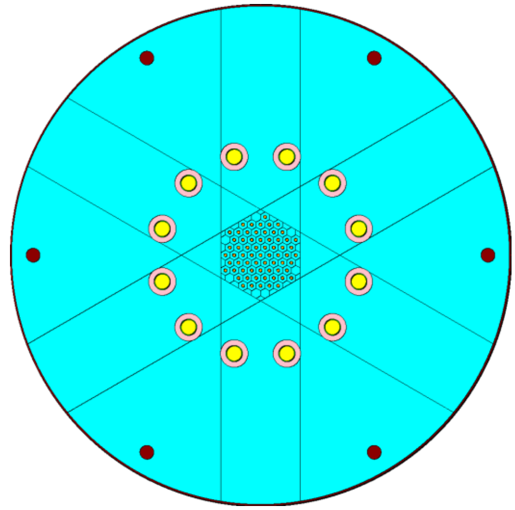

Fig. 13. Horizontal section of SERPENT model 6.

Table 1. Results and parameters of SERPENT (2.1.27) and MCNP (6.1.1β) for the GIACINT critical experimental facility. The statistical error of k_{eff} is reported in pcm units. Both codes simulated 320 million starting fission neutrons per simulation.

Model	Input File [lines]	Computing Time [m]	Parallel Platform	k_{eff}
SERPENT – 1	696	1848.49	OPENMP	1.00057±5.5
SERPENT – 2	703	595.82	OPENMP	1.00067±5.5
SERPENT – 3	1097	600.77	OPENMP	1.00058±5.8
SERPENT – 4	327	570.33	OPENMP	1.00066±6.0
SERPENT – 5	327	179.50	OPENMP	1.00061±5.9
SERPENT – 6	439	167.90	OPENMP	1.00057±5.3
MCNP	563	128.48	MPI	1.00106±5.0

V. CONCLUSIONS

SERPENT has the capability to model a reactor core by STL or hybrid geometry. In the latter case, STL and combinatorial geometries are combined together. The hybrid geometry computation runs much faster than a STL-only geometry computation. An optimized model of the core would use the STL geometry for the complicated parts and combinatorial geometry and lattice for the simple parts.

For the GIACINT facility, SERPENT using the hybrid geometry model runs slower than MCNP using the combinatorial geometry model, since the use of the STL geometry slows down the SERPENT calculation.

With STL geometry, the SERPENT computing time strongly depends on the split levels defined in the *solid* card and *feature angle* parameter that is defined by CUBIT [1].

ACKNOWLEDGMENTS

This work is supported by the U.S. Department of Energy, Office of Material Management and Minimization (M3), National Nuclear Security Administration.

REFERENCES

1. A. TALAMO et al., "Serpent Stereolithography Modeling of the GIACINT Critical Assembly Experimental Facility," *PHYSOR*, Sun Valley, ID, USA, (2016).
2. A. TALAMO et al., Criticality Experiments and Analyses of Uranium Zirconium Carbon Nitride LEU Fuel, *PHYSOR*, Kyoto, Japan, (2014).
3. J. LEPPÄNEN et al., "The Serpent Monte Carlo code: Status, Development and Applications in 2013," *Ann. Nucl. Energy* **82** pp. 142-150 (2015).
4. J. LEPPÄNEN and T. KALTIAISENAHO, "Expanding the Use of SERPENT 2 to Fusion Applications: Shut-Down Dose Rate Calculations," *PHYSOR*, Sun Valley, ID, USA, (2016).
5. 3D SYSTEMS INC., *Stereolithography Interface Specification*, part number 50065-S01-00, (1989).
6. D.B. PELOWITZ et al., "MCNP6 User's Manual, Code Version 6.1.1beta," Los Alamos National Laboratory, LA-CP-14-00745, (2014).
7. CUBIT Toolkit, URL: https://cubit.sandia.gov/, (2016).
8. M.B. CHADWICK et al., "ENDF/B-VII.0: Next Generation Evaluated Nuclear Data Library for Nuclear Science and Technology," *Nuclear Data Sheets* **107**, (2006).