New CAD and unstructured mesh based geometry types in Serpent 2

4th International Serpent UGM, Cambridge, UK, Sept. 17-19, 2014

Jaakko Leppänen
VTT Technical Research Center of Finland

Outline

- Geometry types in transport applications
- New geometry types in Serpent 2:
  1. Unstructured mesh based geometry
  2. Unstructured surface based geometry
- Delta-tracking in complex geometries
- Cell search routine
- Adaptive search mesh
- Example results

NOTE: *Even though the methods presented here are available in Serpent 2, all work is very preliminary*

Background

Serpent was originally developed as a reactor physics code:

- Transport simulation limited to neutrons
- Calculation routines optimized for lattice calculations
- Built-in burnup routine
- Reaction rate tallies calculated using CFE, no variance reduction techniques

Increasing interest to extend the scope of applications to new fields:

- Radiation shielding
- Fusion neutronics
- Medical physics

New applications require development in transport physics (new particle types) and variance reduction, but also in tracking routine and geometry models

Geometry types in transport applications

Constructive Solid Geometry (CSG):

- Homogeneous material cells, formed by combinations of elementary (quadratic) and derived surface types and Euclidean transformations
- Multiple levels (universes), repeated structures created using lattices
- Most common geometry type for Monte Carlo codes, can be used for describing almost any geometry in reactor physics applications

Unstructured mesh based geometries:

- Arbitrary set of points, used for mapping the volume of the 3D object
- Adjacent points are combined to form (planar) facets, which are combined to form homogeneous material cells
- Can be used to describe complex geometries with internal structure
- Used by CFD and structural mechanics codes

Geometry types in transport applications

Unstructured surface based geometries:

- Arbitrary set of points, used for mapping the boundary of the 3D object
- Adjacent points are combined to form (planar) facets, which are combined to form the bounding surface of the solid
- Can be used to describe complex geometries without internal structure
- Used by CAD software

Voxel based geometries:

- Three-dimensional Cartesian mesh, each mesh cell (voxel) assigned with a material
- Used in medical imaging and radiotherapy

Serpent is originally based on the CSG geometry type. The implementation of the other three types is planned, but the work is currently limited to unstructured mesh and surface based geometries.

New geometry types in Serpent 2

Common features to new geometry types:

- Used for modeling complicated irregular structures
- Geometries constructed using 3D design tools or radiographic imaging
- Existing data formats, large files containing non-human-readable data

Why do this with Serpent?

- At the lowest level the geometry routines are still based on the same functions dealing with elementary surface types
- Several routines shared with the multi-physics interface
- Good performance in complex CSG based geometries

The major challenges are related to the size of the models, which affects both memory footprint and computational efficiency

New geometry types in Serpent 2

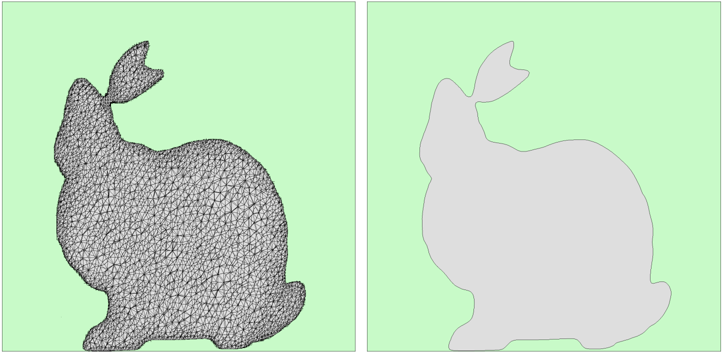


Figure 1: Serpent geometry plot of the “Standford Critical Bunny”, used for testing the new geometry types (high-enriched uranium bunny). Left: unstructured mesh based model with internal structure. Right: unstructured surface based model without internal structure.

New geometry types in Serpent 2

The unstructured mesh and surface based geometries can be used for similar purposes, but there are some differences

Unstructured mesh geometry

- By-product of a multi-physics interface for CFD code coupling
- Based on OpenFOAM mesh and file format
- Internal structure can be used for tallying spatial reaction rate distributions
- Large memory footprint, somewhat error tolerant
- Tested without major problems

Unstructured surface geometry

- Developed separately from multi-physics interface
- Based on STL format, supported by CAD software
- Solid volumes without internal structure
- Smaller memory footprint, faster tracking, no error tolerance whatsoever
- Some issues that need to be resolved

Delta-tracking and complex geometries

The Serpent tracking routine is based on the Woodcock delta-tracking method:

- A rejection sampling based algorithm, that enables particle tracks to be carried over several material boundaries without stopping the track at each surface crossing
- Good performance in geometries where mfp is long compared to dimensions (e.g. HTGR particle fuels)[1]
- Similar advantages in the new geometry types consisting of a large number of small cells and closely-spaced surface – *running time is not expected to be strongly dependent on the resolution of the model*

[1] J. Leppänen. "*Performance of Woodcock Delta-Tracking in Lattice Physics Applications Using the Serpent Monte Carlo Reactor Physics Burnup Calculation Code.*" Ann. Nucl. Energy **37** (2010), 715–722.

Cell search routine

One of the fundamental tasks of the tracking routine is to figure out which material cell is located at position (x, y, z).

For the unstructured mesh based geometry type this task is relatively straightforward:

- Loop over all candidate cells
- Loop over all surfaces comprising the cell
- If the point is inside all surfaces, it is inside the cell

Serpent converts hexa- and other polyhedral meshes into tetrahedral form before the transport simulation is run, so each cell is comprised of four triangular facets.

Cell search routine

The same procedure cannot be applied to unstructured surface based geometries:

- Single solid comprised of a large number of triangular facets
- Complicated geometries are almost always re-entrant

Instead, the cell test works by starting a ray from (x, y, z) into direction (u, v, w). Two optional tests:

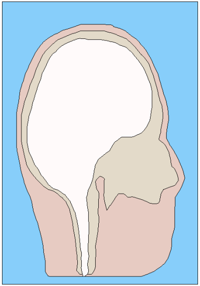
1. If the ray is extended to the first intersection point with the surface, and the scalar product between (u, v, w) and the surface normal yields a positive value (the ray is exiting the volume), the point is inside the solid
2. If the ray is extended to the outer boundary of the geometry, and the surface is crossed odd number of times in between, the point is inside the solid

Test 1 is faster, but more prone to errors (caused by limited precision of floating point arithmetics?)

Adaptive search mesh

Unstructured mesh and surface based geometries may consists of well over 100,000 triangular facets. Performing the tests for all cells or facets is not a practical way to perform cell search. Instead, the routine is based on an adaptive search mesh:

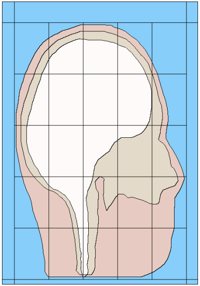
- Instead of looping over all cells or facets, the tests are limited inside the local search mesh cell
- The finer the mesh, the faster the search
- Instead of using a simple Cartesian mesh, the structure is adaptively refined where higher resolution is needed
- Leads to considerable speed-up in the tracking routine
- Material information can be pre-assigned to empty search mesh cells in surface based geometries



Adaptive search mesh

Unstructured mesh and surface based geometries may consists of well over 100,000 triangular facets. Performing the tests for all cells or facets is not a practical way to perform cell search. Instead, the routine is based on an adaptive search mesh:

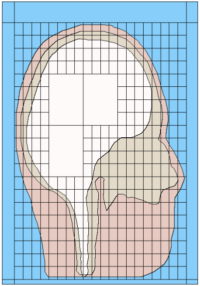
- Instead of looping over all cells or facets, the tests are limited inside the local search mesh cell
- The finer the mesh, the faster the search
- Instead of using a simple Cartesian mesh, the structure is adaptively refined where higher resolution is needed
- Leads to considerable speed-up in the tracking routine
- Material information can be pre-assigned to empty search mesh cells in surface based geometries



Adaptive search mesh

Unstructured mesh and surface based geometries may consists of well over 100,000 triangular facets. Performing the tests for all cells or facets is not a practical way to perform cell search. Instead, the routine is based on an adaptive search mesh:

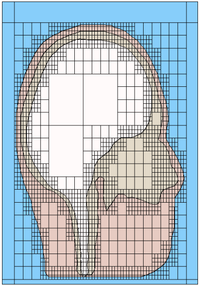
- Instead of looping over all cells or facets, the tests are limited inside the local search mesh cell
- The finer the mesh, the faster the search
- Instead of using a simple Cartesian mesh, the structure is adaptively refined where higher resolution is needed
- Leads to considerable speed-up in the tracking routine
- Material information can be pre-assigned to empty search mesh cells in surface based geometries



Adaptive search mesh

Unstructured mesh and surface based geometries may consists of well over 100,000 triangular facets. Performing the tests for all cells or facets is not a practical way to perform cell search. Instead, the routine is based on an adaptive search mesh:

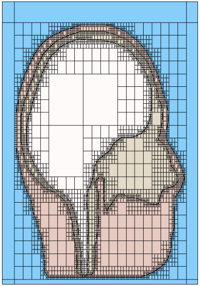
- Instead of looping over all cells or facets, the tests are limited inside the local search mesh cell
- The finer the mesh, the faster the search
- Instead of using a simple Cartesian mesh, the structure is adaptively refined where higher resolution is needed
- Leads to considerable speed-up in the tracking routine
- Material information can be pre-assigned to empty search mesh cells in surface based geometries



Adaptive search mesh

Unstructured mesh and surface based geometries may consists of well over 100,000 triangular facets. Performing the tests for all cells or facets is not a practical way to perform cell search. Instead, the routine is based on an adaptive search mesh:

- Instead of looping over all cells or facets, the tests are limited inside the local search mesh cell
- The finer the mesh, the faster the search
- Instead of using a simple Cartesian mesh, the structure is adaptively refined where higher resolution is needed
- Leads to considerable speed-up in the tracking routine
- Material information can be pre-assigned to empty search mesh cells in surface based geometries



Example results

The Unstructured mesh and surface based models have been tested using various toy problems:

- Papers in PHYSOR 2014[2] and ANS Winter Meeting 2014[3]
- Calculations using a mesh based model show that the running time is not strongly dependent on the mesh size
- Surface based model is slightly faster than the mesh based model, and both compare well to a simplified CSG model (factor of 1.5-2.0 between Stanford Critical Bunny and Godiva)
- Use of delta-tracking results in a considerable speed-up, especially in mesh based models with internal structure
- Adaptive search mesh is an efficient way to reduce the running time
- Mesh based model has much larger memory footprint than the surface based model

---

[2] J. Leppänen and M. Aufiero. "*Development of an Unstructured Mesh Based Geometry Model in the Serpent 2 Monte Carlo Code.*" In proc. PHYSOR 2014. Kyoto, Japan, Sept. 28 - Oct. 3, 2014.

[3] J. Leppänen. "*Development of a CAD Based Geometry Model in Serpent 2 Monte Carlo Code.*" In proc. ANS Winter Meeting 2014. Anaheim, CA, Nov. 9 - 13, 2014.

Example results

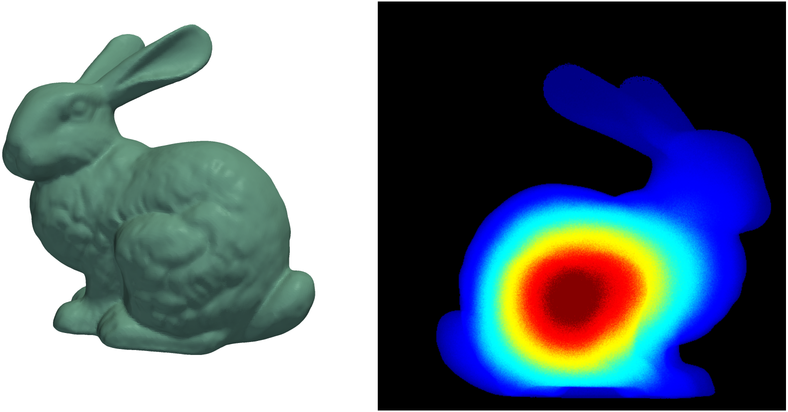


Figure 2: The “Stanford Critical Bunny”. Left: 3D geometry model. Right: Fission rate distribution calculated by Serpent.

Example results

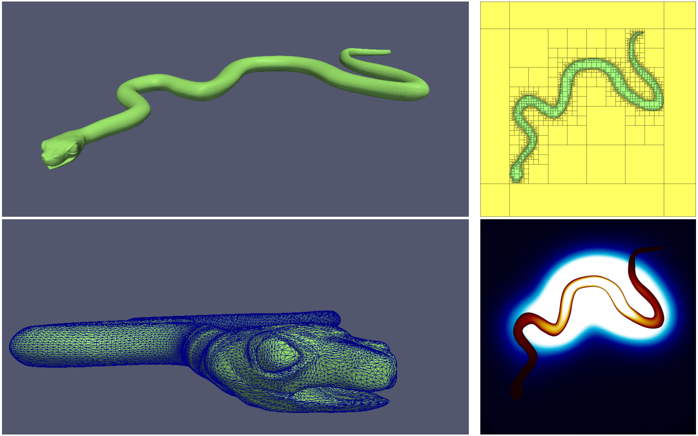


Figure 3: A prompt super-critical plutonium snake in water (STL model)

Example results

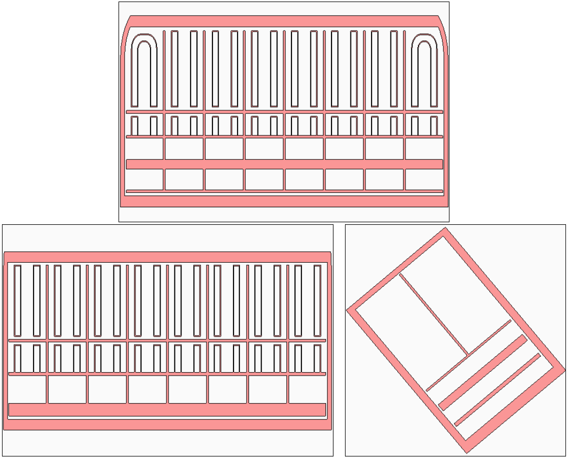


Figure 4: Cross-sectional views of an ITER test blanket module (TBM) used for tritium breeding. STL format geometry model consisting of 2270 points and 4472 triangular facets.

Potential applications

The first intended application for the new 3D geometry types is fusion neutronics:

- CAD models for JET, ITER and DEMO exist
- Serpent has a built-in burnup routine, which can be used for tritium breeding and material activation analyses
- Material damage calculations (DPA) possible using detectors

Plans for future:

- Coupling Serpent with Apros system code at VTT:
  - Enabling Research project proposal *“Fusion technology extenstion for the Apros process simulation platform”* submitted to EUROfusion WP2015
  - Contact: Dr. Markus Airila from VTT (markus.airila@vtt.fi)
- Collaboration with Aalto University:
  - Possibility for student exchange with Aalto
  - Contact: Professor Mathias Groth from Aalto University (mathias.groth@aalto.fi)

Summary and conclusions

- Unstructured mesh and surface based geometry models have been implemented in Serpent 2
- The intended purpose is to provide an efficient way to model complicated geometries encountered in new applications, such as radiation shielding or fusion neutronics
- The geometry data is imported directly to Serpent without conversion to CSG, in the form of OpenFOAM mesh or STL surface files produced by 3D design tools
- The preliminary results using various toy problems have been promising, but more experience is needed


TECHNOLOGY FOR BUSINESS