Layered CAD/CSG Geometry for Neutronics Modeling of Advanced Reactors

Elliott Biondo, Gregory Davidson, Brian Ade
Oak Ridge National Laboratory, 1 Bethel Valley Rd., Oak Ridge, TN, 37830

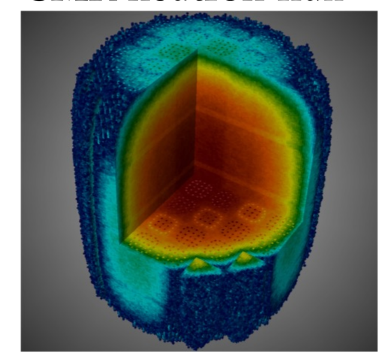
Shift Overview

- ORNL's flagship Monte Carlo (MC) radiation transport code
- Fission, fusion, and defense applications

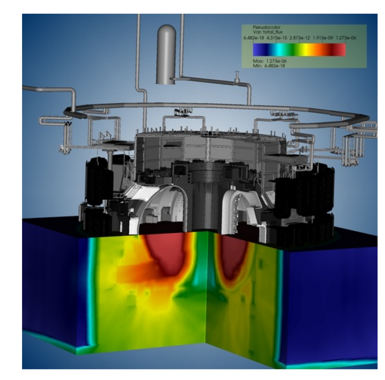
Features

- Fixed-source and k-eigenvalue simulations
- Designed to scale from laptops to supercomputers with CPU and GPU execution
- Supports Shift CSG, "reactor aware" CSG, MCNP CSG, and DAGMC CAD geometry
- Automatic coupling with the Denovo 3D S_N code to perform CADIS/FW-CADIS
- Advanced features such as non-uniform domain decomposition, inline depletion, and on-the-fly Doppler broadening

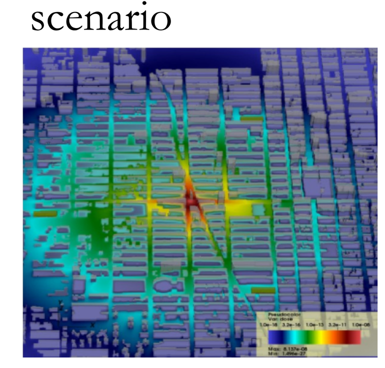
SMR neutron flux



ITER neutron flux



Dose rate, urban scenario



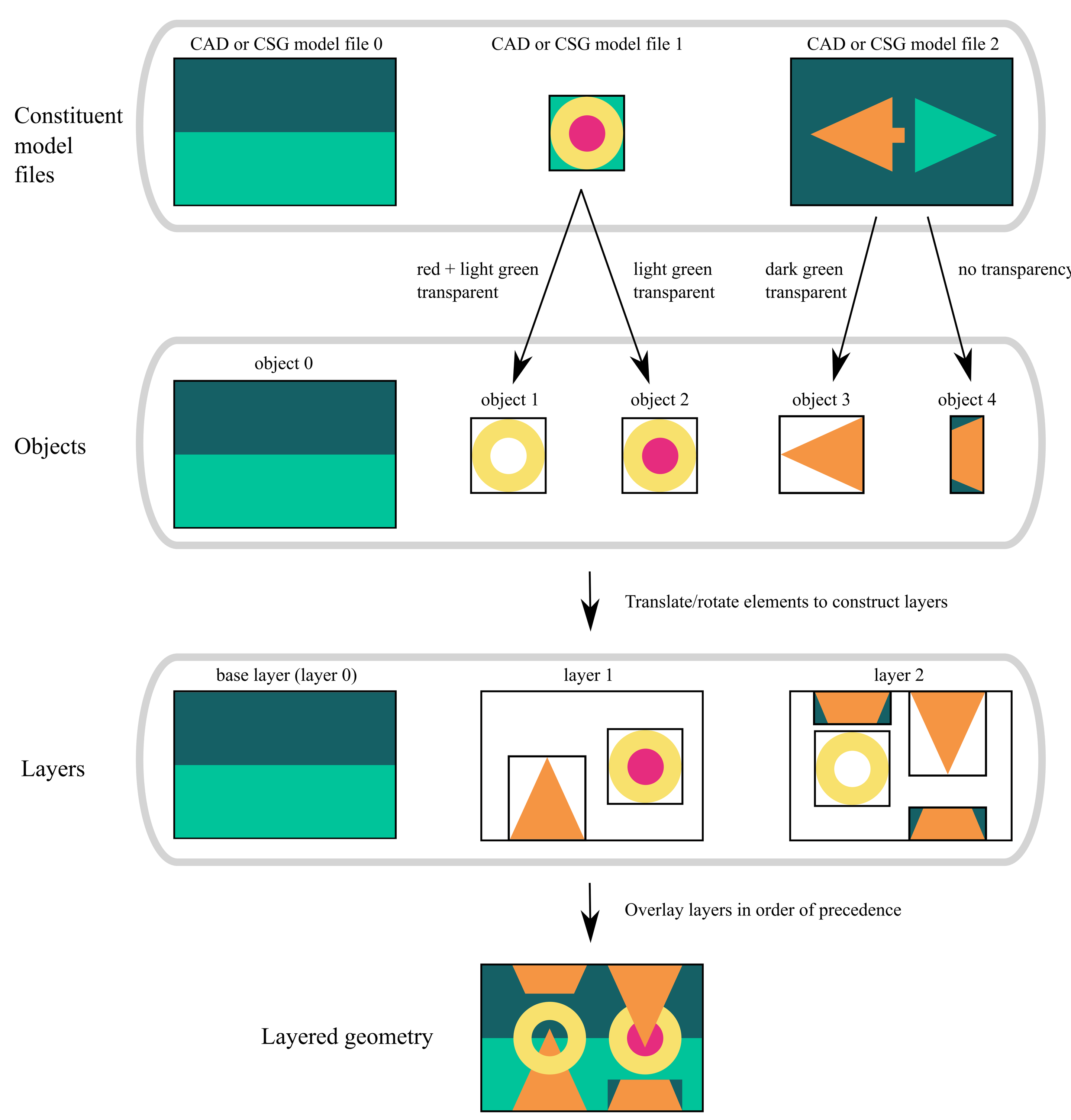
CSG and CAD Geometries

Constructive Solid Geometry (CSG)	Computer-Aided Design (CAD)
• Primitive shapes combined using logic operations • Generally linear or quadratic surfaces • Tedious to create/combine	• Models converted into surface/volume meshes • Support for higher order surfaces • Higher RAM requirements

- Combine CSG/CAD for advanced reactor analysis:
  - Use existing CAD models from thermal hydraulics or structural mechanics analysis
  - Rigorously model complex components such as spacer grids or additively manufactured fuel elements

Layered Geometries in Shift

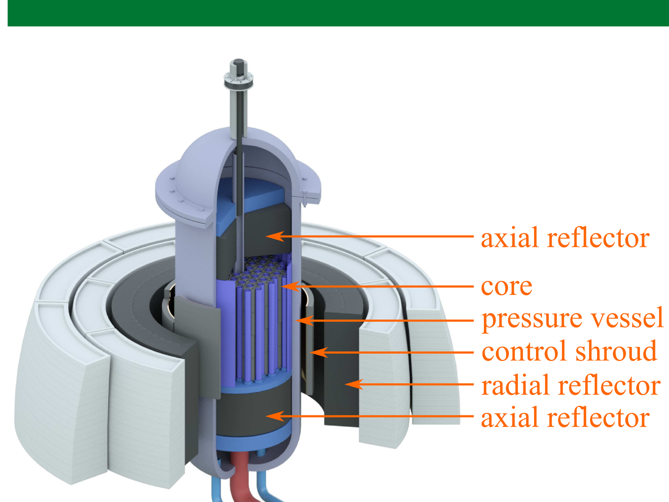
- A *layered* geometry type has been implemented in Shift, allowing CSG and CAD geometries to be combined arbitrarily
- Constructed from *constituent models*, *objects*, and *layers*
- Tracking is done on all layers independently, with cross sections and tallies evaluated only on the *active* layer
- Standard MC tracking algorithm left intact, with layered geometry modifications to the following routines:
  - distance_to_boundary
  - move_across_surface
  - move_within_cell




```
procedure FIND_ACTIVE_LAYER(pos)
    for layer_index ∈ [max_layer_index ... 0]
        for object ∈ layers[layer_index].objects()
            if object.contains_point(pos)
                cell ← object.find_cell(pos)
                if not cell.material.is_transparent()
                    return layer_index
                end if
            end if
        end for
    end for
end procedure
```

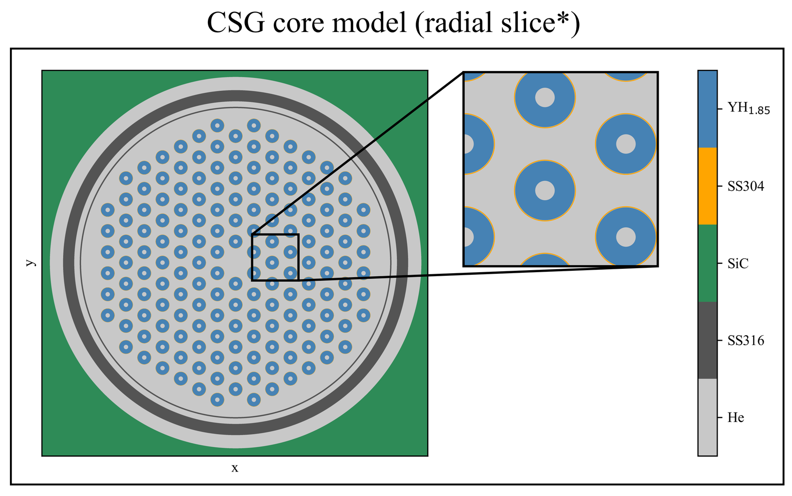
```
procedure DISTANCE_TO_BOUNDARY(pos, dir)
    min_dist = distance_to_outer_boundary(pos, dir)
    for layer ∈ layers
        current_obj = layer.find_object(pos)
        if current_obj is not null
            obj_dist ← current_obj.distance_to_boundary(pos, dist)
            min_dist = min(min_dist, obj_dist)
        else
            for obj ∈ layer.objects()
                bbox_dist = distance_to_box(obj.bounding_box, pos, dir)
                min_dist ← min(min_dist, bbox_dist)
            end for
        end if
    end for
    return min_dist
end procedure
```

Transformational Challenge Reactor (TCR) Demonstration Problem

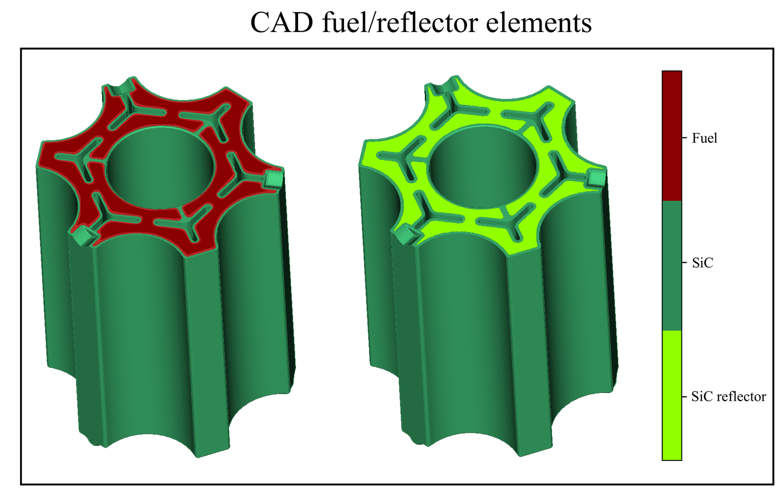


- Conceptual reactor design with a core consisting of additively manufactured cog-shaped fuel elements
- Fuel elements were designed in CAD and contain higher order surfaces not easily represented in CSG

CSG core model (radial slice*)



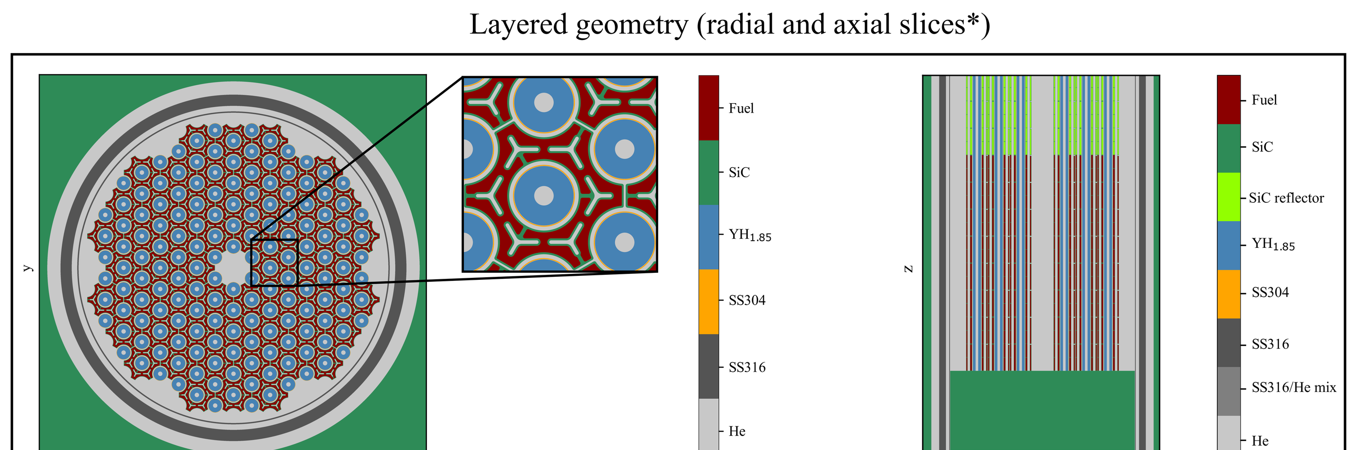
CAD fuel/reflector elements



Base layer

594 elements arranged in a hexagonal grid of assemblies

Layered geometry (radial and axial slices*)



*Radial reflector truncated to show detail

- Cold zero power simulation with 25 inactive cycles and 25 active cycles, with 5×10^6 histories per cycle on 960 CPU cores
- Flux, fission source, and k_{eff} results matched expectations