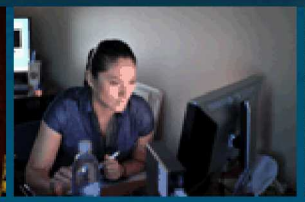
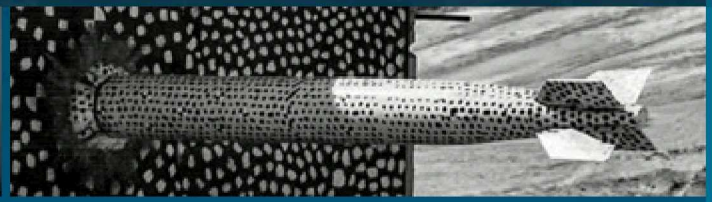
SAND2019-11473PE

Generic Model Implementation for GADRAS





Presented By

Kyle Kercher

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Abstract

The goal of this project is to utilize the preexisting framework of GADRAS to simulate the radiation leakage from arbitrary CAD models without sacrificing speed or accuracy. The proposed solution is to use STL files to define the model. Then, create a 3 dimensional binning structure to contain all of the elements of the file. This results in preservation of speed, without adding hardware requirements. Finally, the discretization is performed using a 3 dimensional framework to utilize GADRAS' refinement algorithm. The combination of these two ideas, result in an absolute error within 10% for standard conditions, and 20% for edge case conditions. The addition of arbitrary models will simplify the modeling process for complex shapes, allow for more flexible models, and allow for creation of models that are simply impossible in the current framework.

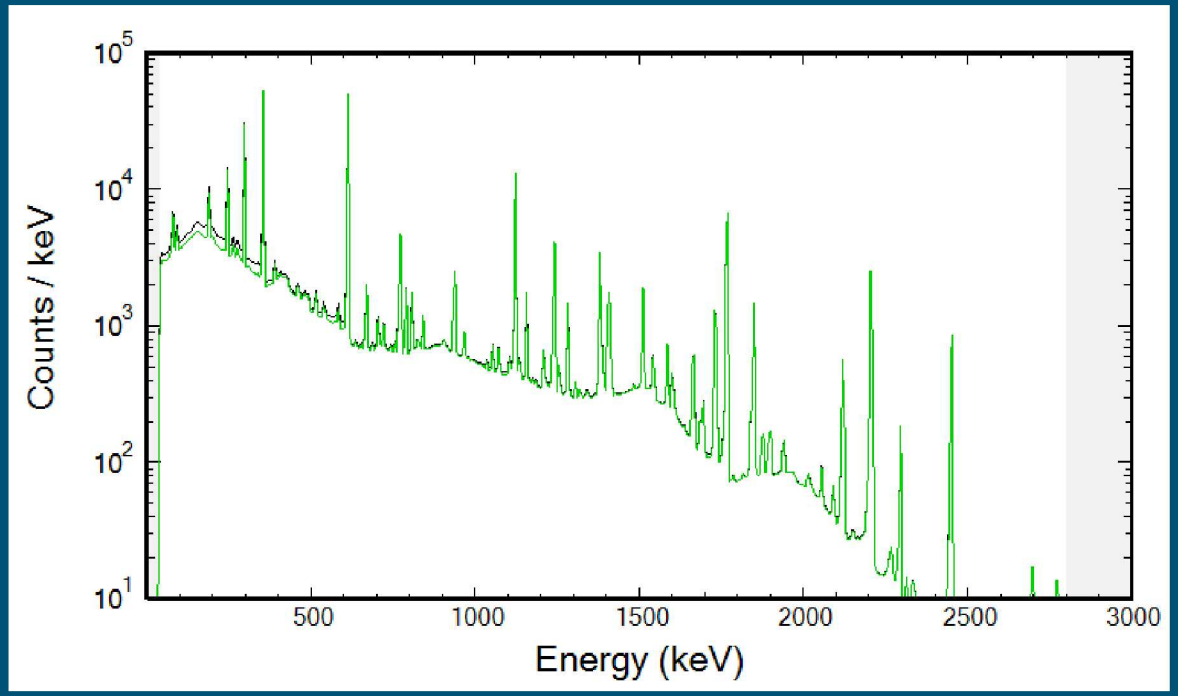
Table of Contents

What is GADRAS?

- Gamma Detector Response and Analysis Software
  - Model gamma and neutron detector response
  - Characterize detectors
  - Analyze spectra for isotope identification
  - Display measured and computed spectra
  - Simulate spectra from 1D or 3D source models

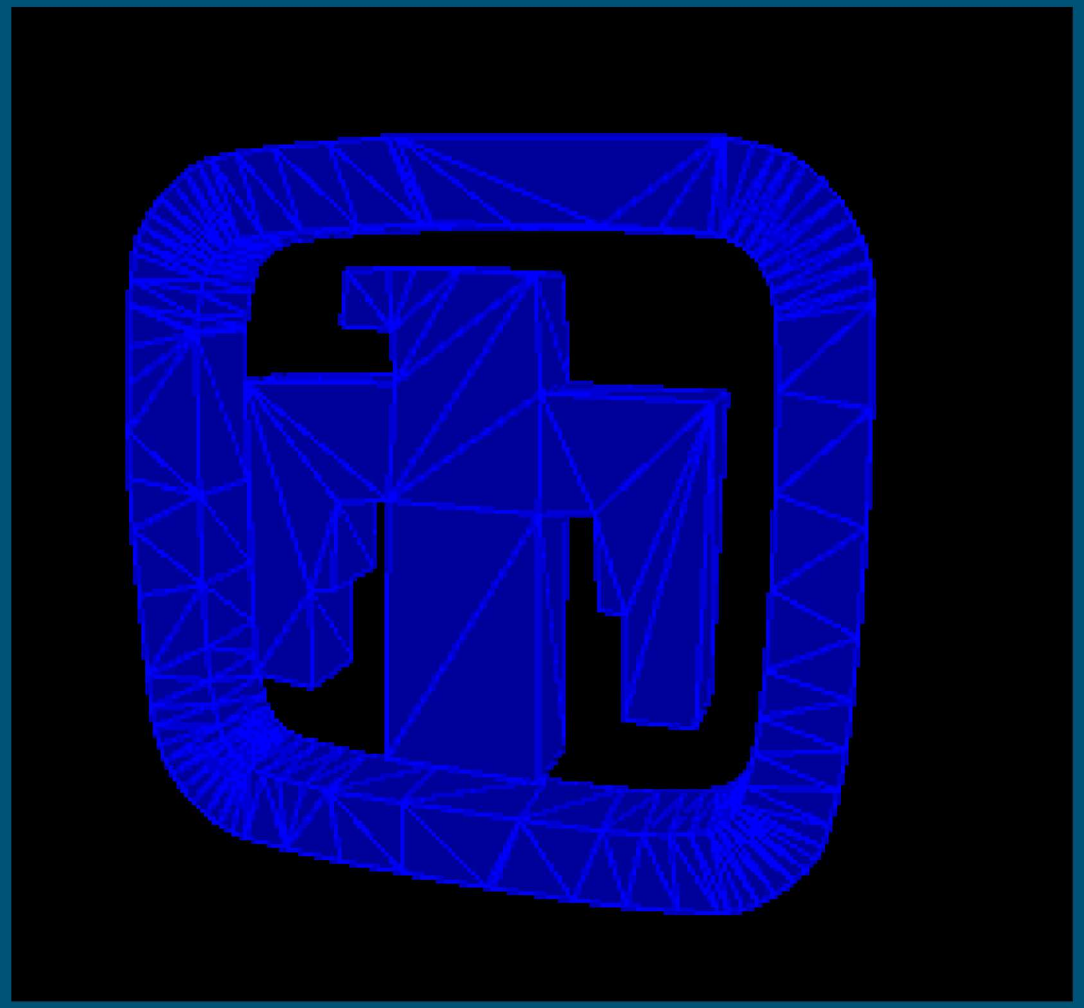
GADRAS Solution Method

- Build Scene using shape primitives
  - Spheres, Cubes, Cylinders, Cones, Caps, Round Cylinders
- Discretize scene into "point" sources
  - Create voxels for specific geometries
- Solve contribution for each voxel
  - Ray trace through scene for attenuation
  - Use materials along ray to estimate scatter
- Combine for total detector response



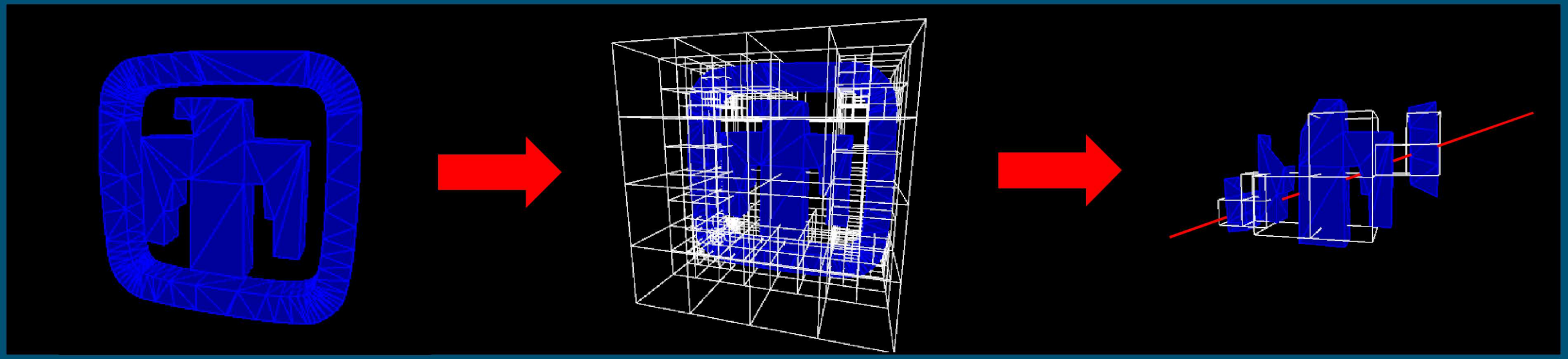
File Selection

- SolidWorks CAD file
  - Not standardized
- Generic STP file
  - 100's of different operations
- IGES file
  - ~50 different operations
- STL File
  - Large number of elements

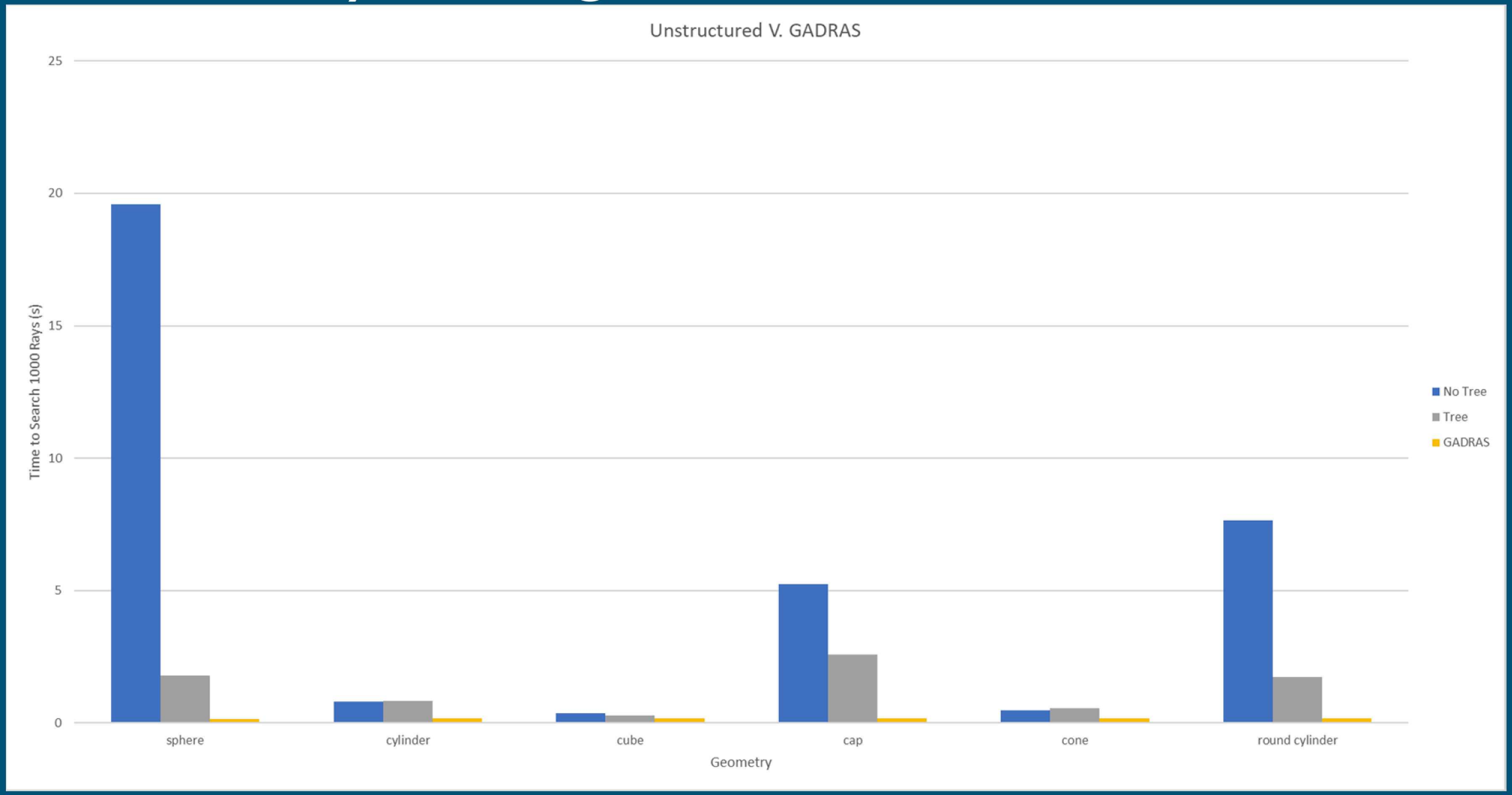


Octree Optimization

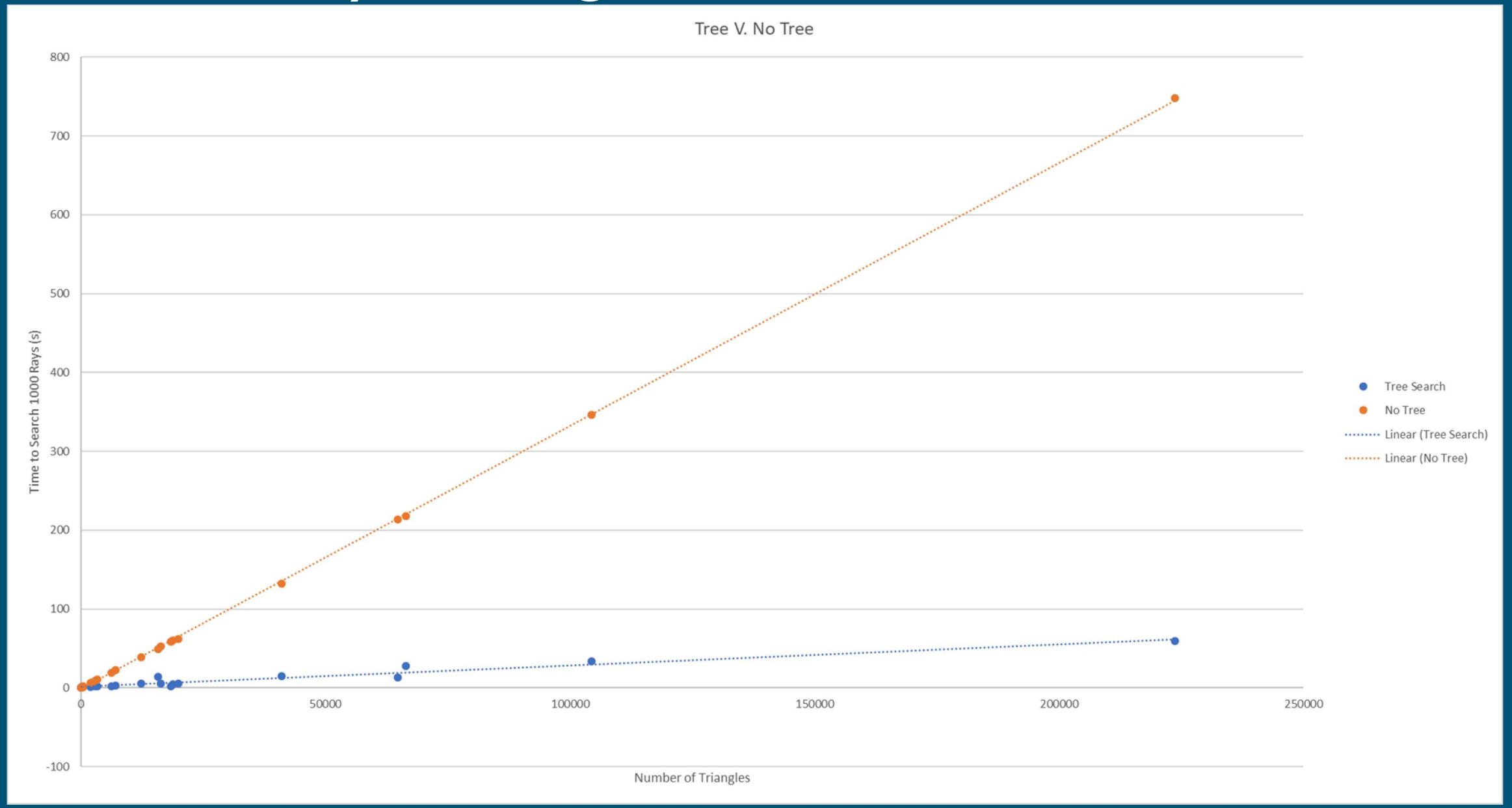
- Recursive 3D binning data structure
  - Each node can divide into 8 children nodes
- Reduce the search space for each ray
  - 10K triangles to 200 triangles



Results: Ray Tracing



Results: Ray Tracing

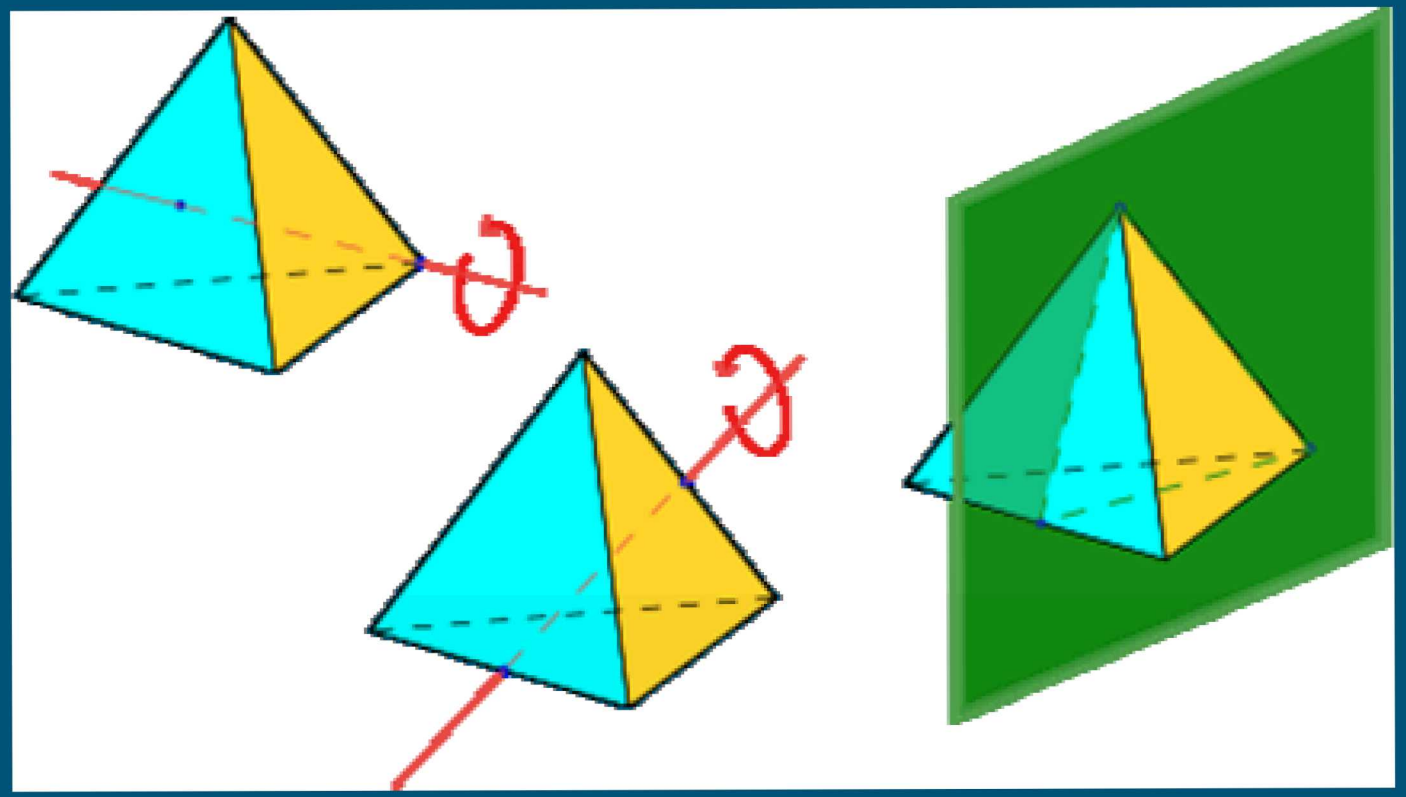


GADRAS Voxelization

- Defines each primitive with 3 dimensions
  - E.g. Spheres: Radius, Theta and Azimuth
- Create an initial sparse mesh
- Calculate importance of voxels
  - Based on volume, and overall contribution
- Refine based on 5 criteria:
  - Other models
  - External Attenuation
  - Self Attenuation
  - Geometric Attenuation
  - Source term gradient

Division of Tetrahedron

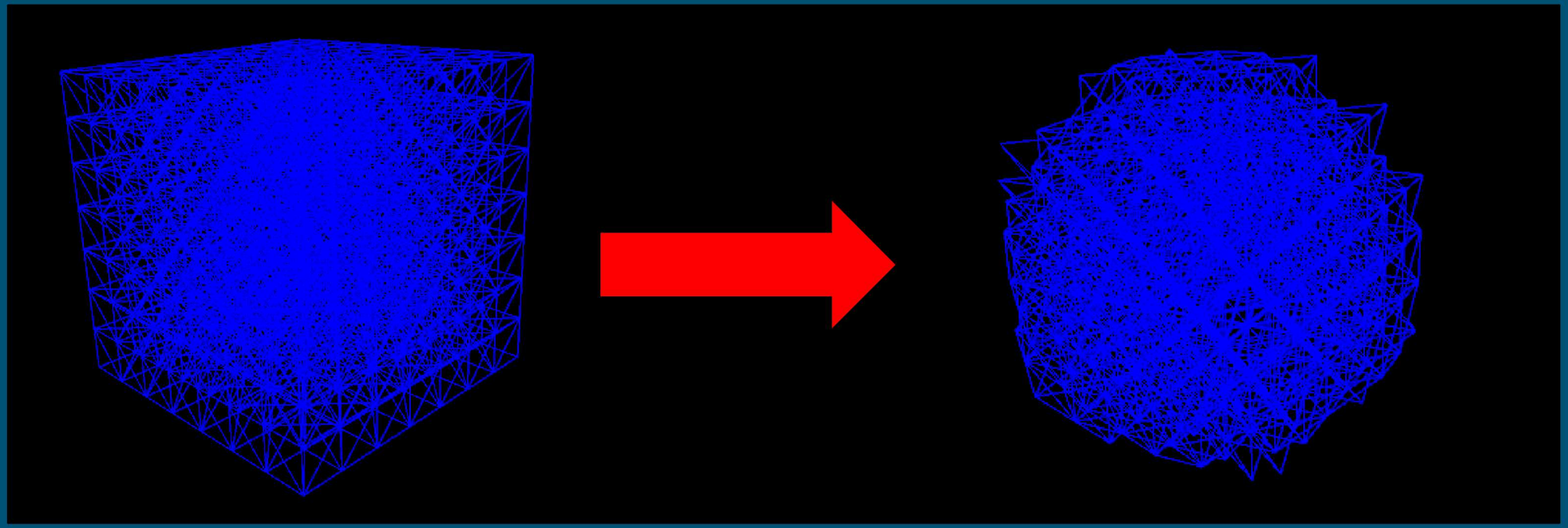
- Using one edge and a point on the opposite edge, create two new elements



[1]

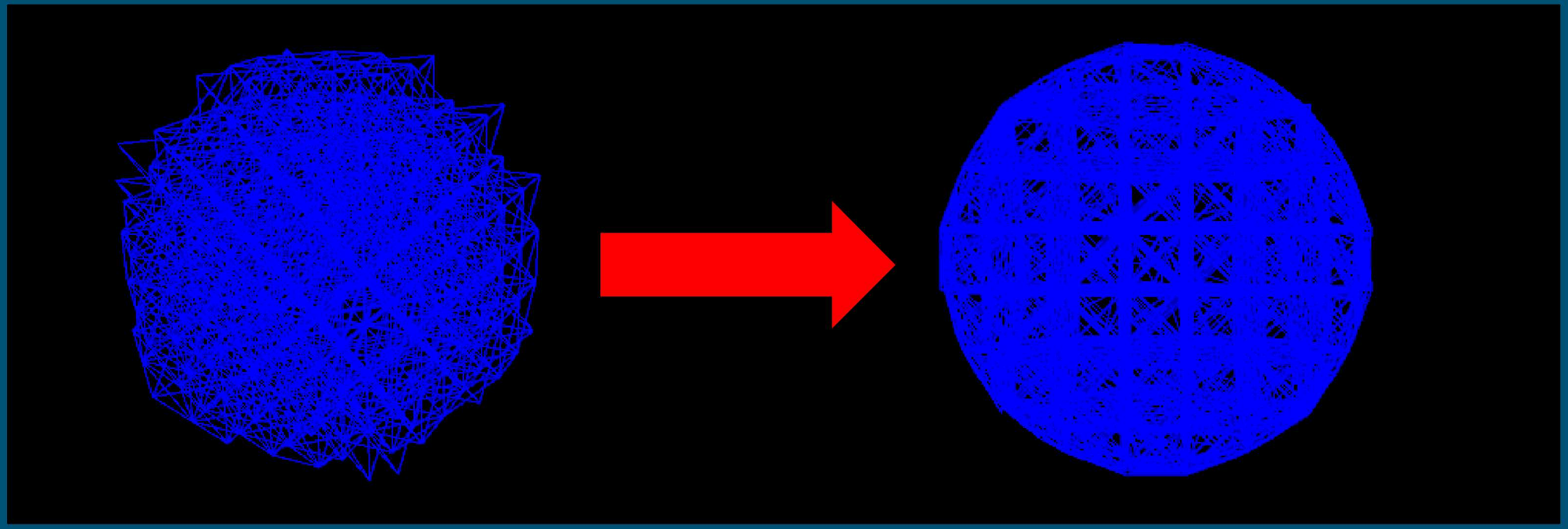
Initial Mesh

- Populate the volume of the model with cubes
- Divide each cube into 5 tetrahedron
- Remove any tetrahedron not in the model



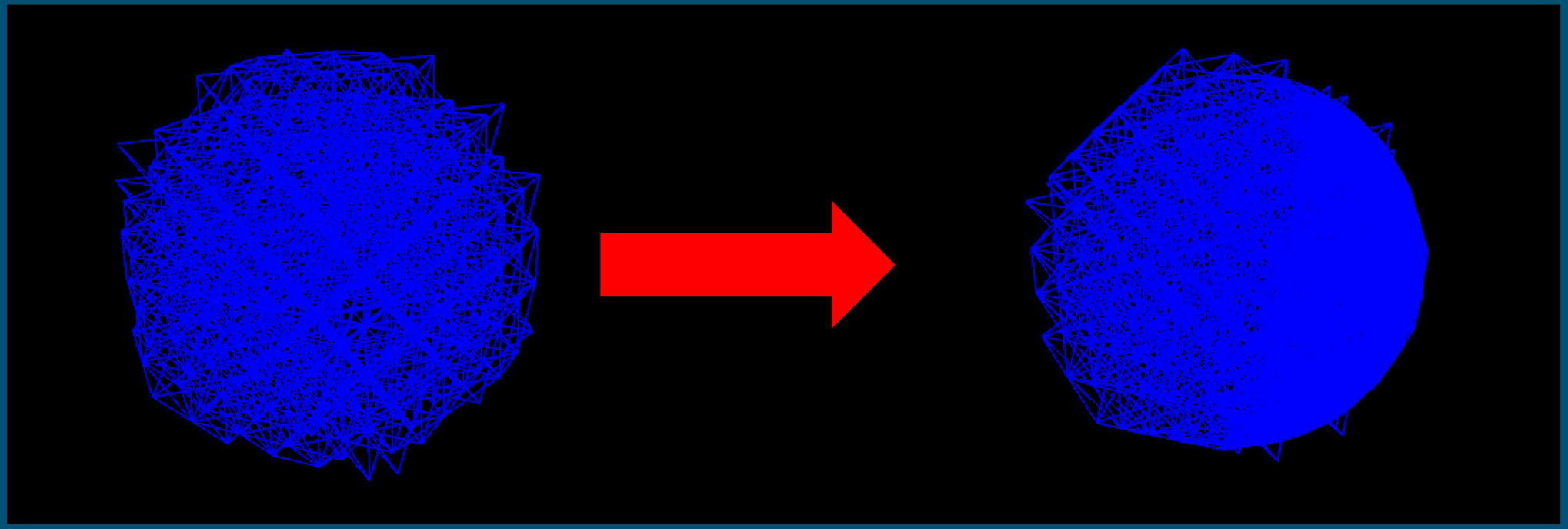
Model Refinement

- Ray Trace along edges of Tetrahedron on surface of model
- Create two tetrahedron with the split point located on the surface
- Refine until geometric tolerance or max voxels are met

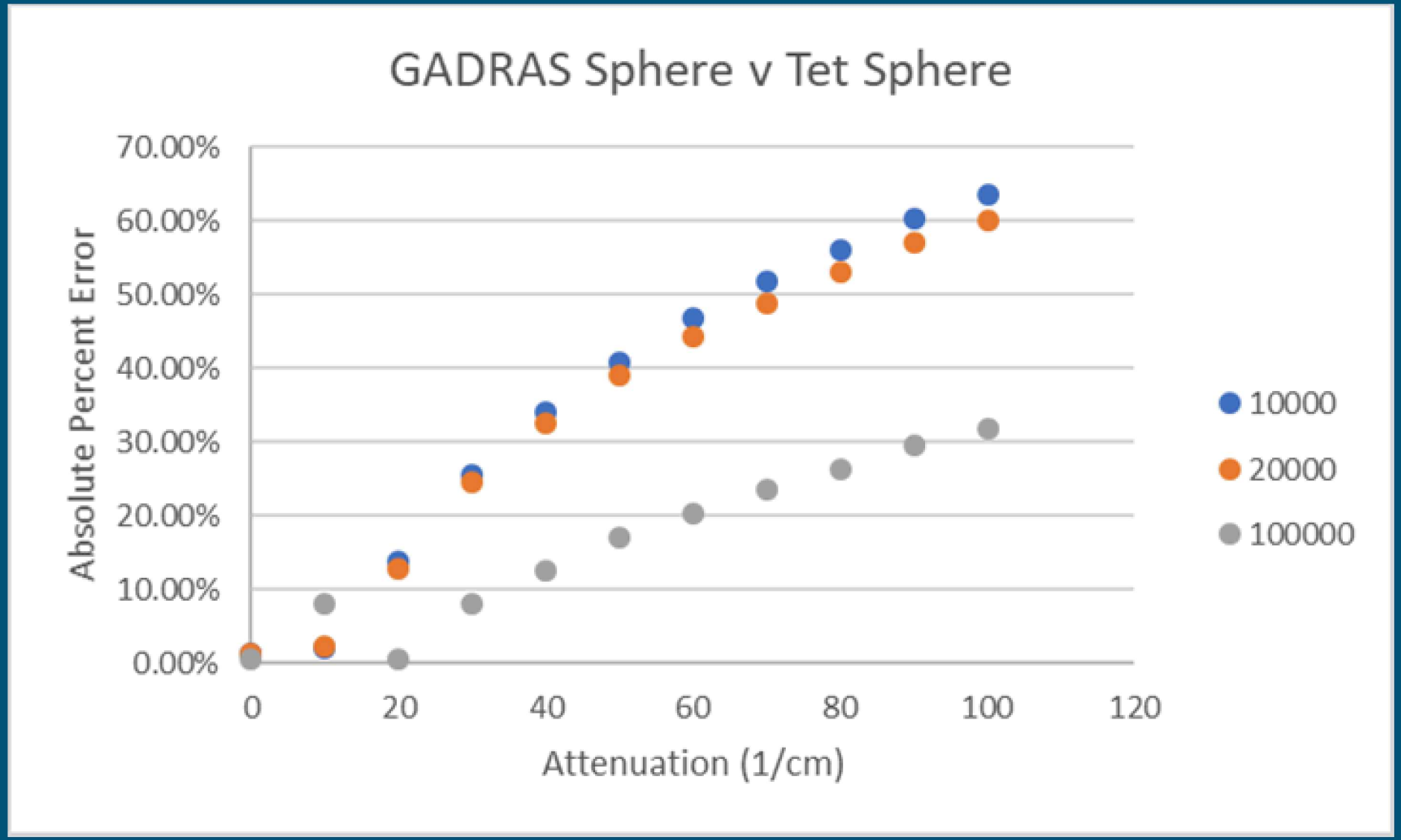


3 Dimension Refinement

- Each dimension is a cross product of opposite edges
- Used for the other 4 refinement requirements



Results: Voxelization



Future Work

- Currently restricted to simple source terms
- Large number of voxels needed for good convergence
- Optimization of local parameters needed for general case

Questions?

Sources

[1] https://en.wikipedia.org/wiki/Tetrahedron