NSTX-U Neutron Flux VR Visualization

Jamal Johnson (*SULI '19*)

Abstract

The goal of this work was to develop a 3 dimensional VR visualization technique for neutron flux (abbreviated *n-flux*) throughout the room housing the National Spherical Torus Experiment Upgrade (NSTX-U) tokamak scheme fusion reactor. The primary application of which is the visual identification of n-flux location boundaries in a virtual 1:1 scale representation of the NSTX-U room where sensitive electronic components are most at risk from high intensity neutrons resonant in energy with materials such as silicon-28. Atilla, a radiation modeling program, was used to simulate neutron transport from the NSTX-U throughout an empty room and provide flux with location grid points. Utilizing that data, the marching cubes algorithm was applied in Unity to create the visualization. This technique ultimately constructs a single valued surface (or *isosurface*) through a volume of space. Grid data of high-risk flux for neutron energies resonant with silicon 28 nuclei were implemented and inspected in VR.

NSTX-U Neutron Flux VR Visualization

Jamal Johnson (*SULI '19*)

Advised by Jonathan Klabacha (*Princeton Plasma Physics Lab*)

Abstract // Introduction

The goal of this work was to develop a VR visualization technique for neutron flux throughout the room housing the National Spherical Torus Experiment Upgrade (NSTX-U) tokamak. This would identify boundaries in a virtual 1:1 scale room where electronics are most at risk from intense, resonant neutron flux. Atilla, a radiation modeling program, was used to simulate neutron transport from the NSTX-U throughout the room and provide flux with location data. Utilizing that data, the Marching Cubes algorithm, a technique that constructs a single valued surface (or *isosurface*) through a volume, was applied in Unity. Data for neutrons resonant with ^{28}Si were then inspected in VR.

LBTE // Atilla

The Linear Boltzmann Transport Equation (LBTE) is the most important equation in neutron transport. It represents the balance of particles over a control volume element dV. This incorporates the unit vector of the particle direction $\widehat{\Omega}$, the total macroscopic cross-section of the interaction σ_t $(length^{-1})$ and the directional derivative operator $\widehat{\Omega} \cdot \nabla$ to relate the position $\vec{r}$, angle $\widehat{\Omega}$ and energy E dependent particle flux function $\psi(\vec{r}, \widehat{\Omega}, E)$ to the summed contributions of external Q_{ext} and scattering Q_{scat} particle sources. [3]

$$\underbrace{\widehat{\Omega} \cdot \vec{\nabla}\psi}_{\text{streaming}} + \underbrace{\sigma_t \psi}_{\text{collision}} = \underbrace{Q_{scat} + Q_{ext}}_{\text{sources}}$$

The streaming operator gives the quantity of particles entering dV, while the collision operator gives the quantity leaving the volume element. [3]

Atilla solves the LBTE via discretization of the equation in space, angle and energy. It constructs tetrahedral volume elements then iterates within each element until convergence (when the value of interest stops significantly changing) is reached. [3]

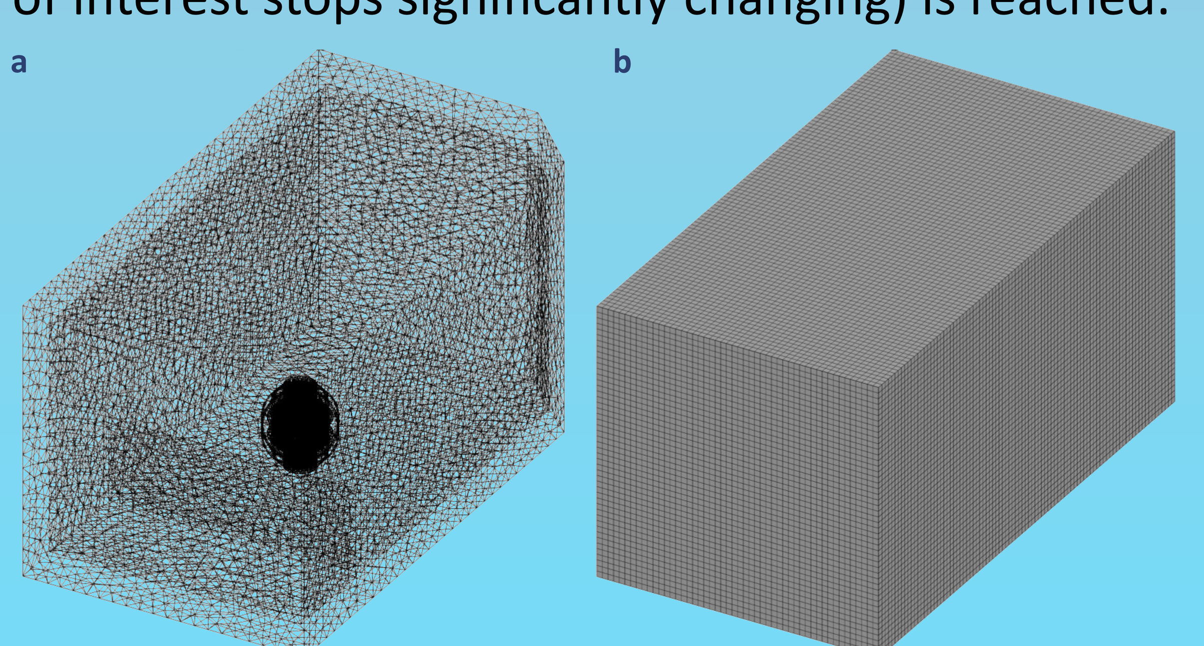

Fig. 1 | Visualized Atilla Solution Space. a. A discretized mesh of unstructured tetrahedrons are constructed for the NSTX-U room for element wise iteration. **b.** Solutions are extrapolated to a structured cubic element grid. Flux values have now been calculated for each vertex position.

After specifying an energy group, the solution for all defined space specific to that group may be calculated. Referenced in figure 2, the 0.055 to 7.408 MeV neutron group was selected and its solution data obtained.

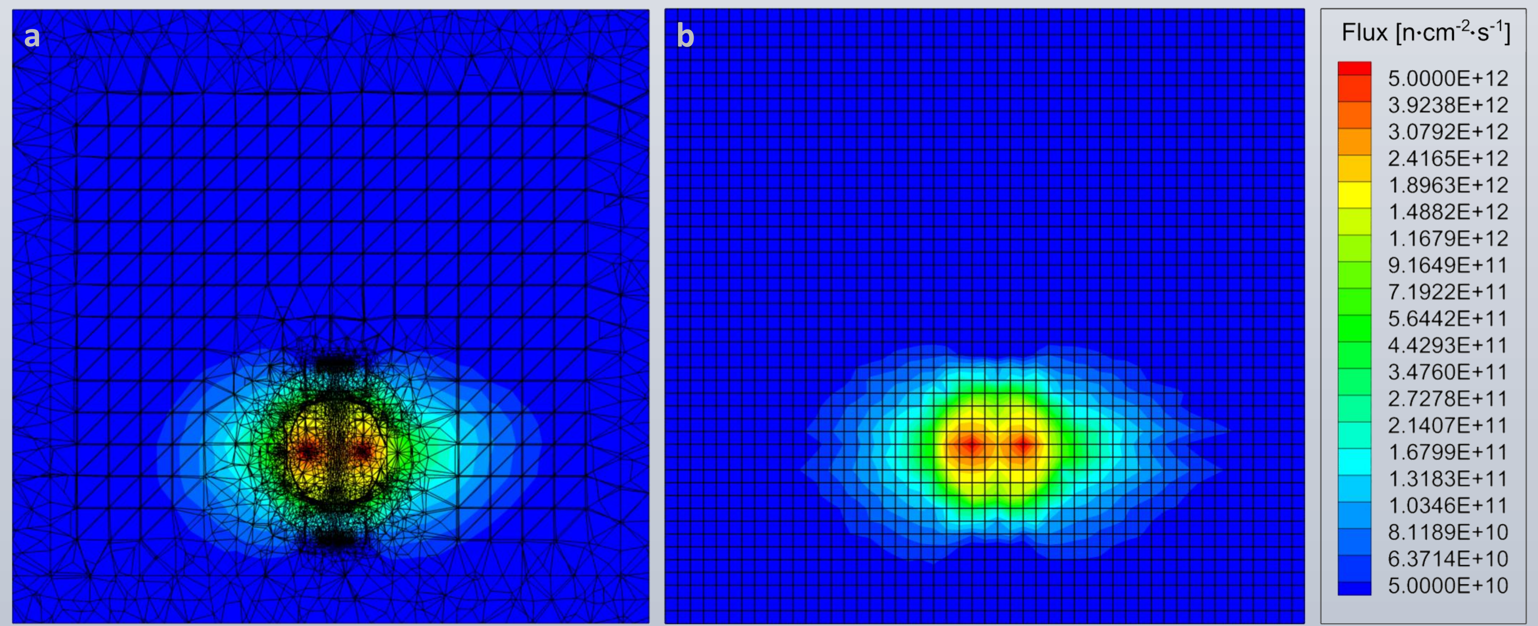

Fig. 2 | Flux Isosurfaces for $^{28}Si(n,X)Y$ Resonance Energy Bin. a. A cross-section of the tetrahedron mesh shows flux values for that plane of space. **b.** The same plane is shown after extrapolating the volume to a cubic structure of data point locations. Resolution about the walls of the NSTX has decreased due to the simplified mesh and increased for other areas. The full grid contains 229,910 data points.

Shown in figure 3, this neutron energy bin spans the resonance energy range for ^{28}Si. Thus the probability of neutron interaction with silicon nuclei becomes much greater and electrical components containing ^{28}Si should be kept out of range of high intensity regions.

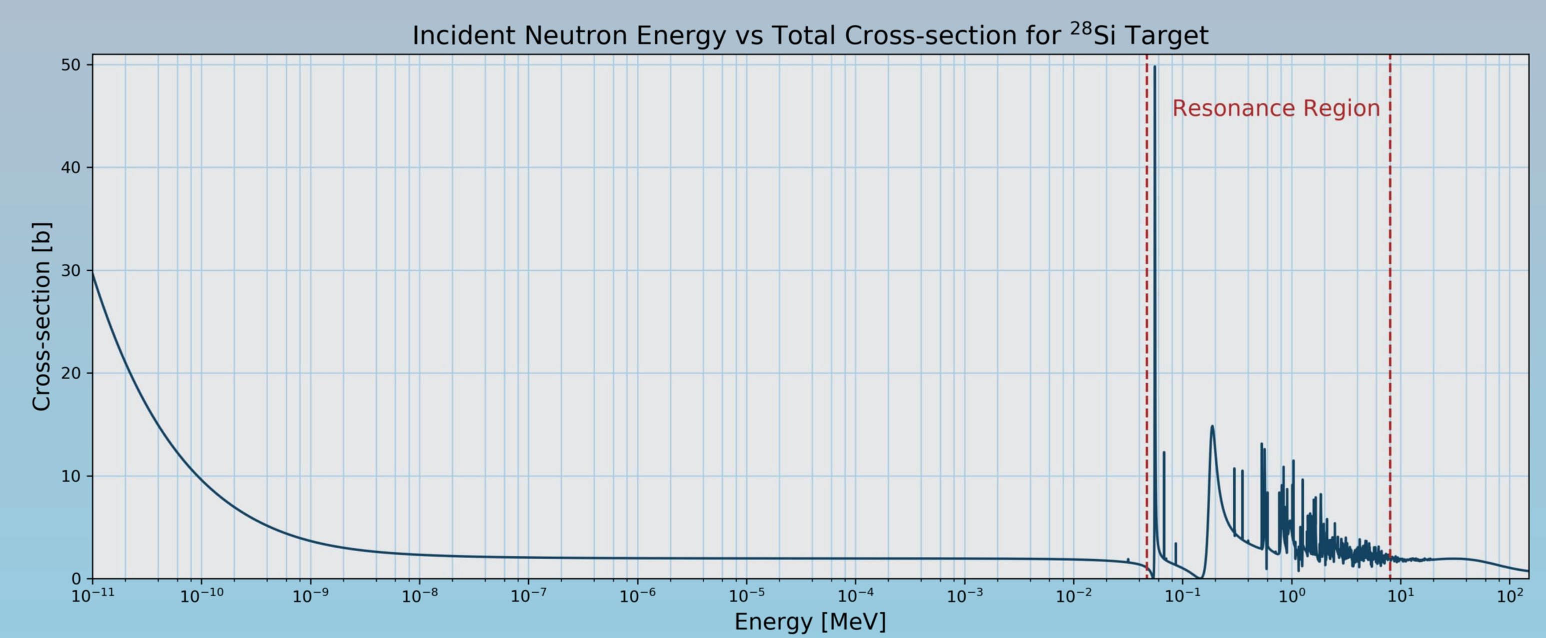

Fig. 3 | Incident Neutron Energy and Total Cross-section for ^{28}Si. Though the interaction cross-section of the target material decays exponentially with increasing neutron kinetic energy, the cross-section becomes large when neutron and target nuclei are in resonance. This greatly raises the probability of inelastic collision and is a significant threat to ^{28}Si electronic components. [4]

Marching Cubes // Unity

The Marching Cubes algorithm produces a 3D contour of a scalar field by approximating facets to an isosurface. A rectangular structured grid where the values of all points are known is required. Single grid cells (cubes) are defined by the values at each vertex. Each cube acquires a byte number after each of the values at its vertices are used to assemble an 8 bit sequence. If a given vertex value is equal to the specified isosurface value (or its value is *under* the surface's) it is considered active and a 1 is placed into the bit sequence. If the vertex value is neither on or within the surface, a 0 is placed instead. [1]

Once complete, a byte number is obtained and used as the index number in a triangulation table containing the edge indices cut within that cube. There are 256 possible cases (byte numbers ranging from 0 - 255) that a cube may receive. [1]

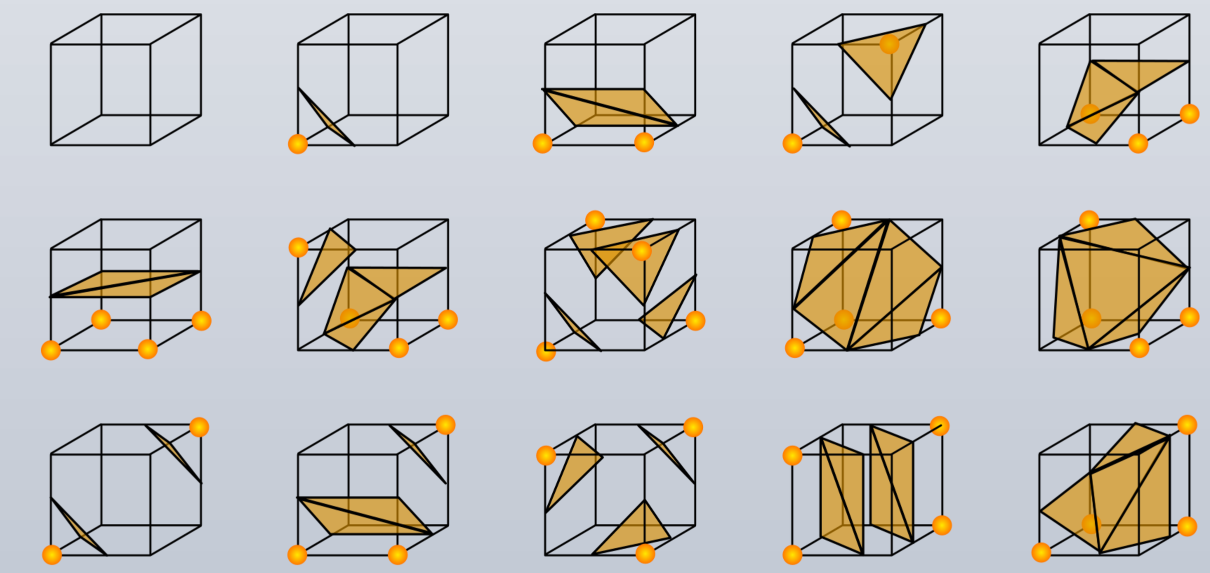

Fig. 4 | Unique Cube Cases. Though facets of only 15 cases are shown, the remaining 241 cases may be expressed as symmetries of those displayed. Active vertices are marked with a bright orange circle. [2]

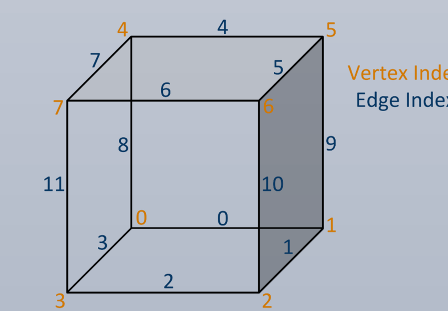

Fig. 5 | Cube Indexing Scheme. To work with the most common triangulation table, vertex and edge indices of each cube must be designated as shown. The sequence of integers representing edge indices in each triangulation table entry must be followed when rendering the facets. Facets are produced by triplets of vertices from the isosurface cutting those edges. Triangles are drawn by connecting vertices in a clockwise manner and only visible from that side's perspective. [1]

After identifying cut edges, a vertex positioned at where the iso-value would lie is interpolated for each edge using the values and locations of the vertices on the ends of that edge. [1]

Isosurface vertices and their drawing (or connecting) sequence are then stored in the mesh filter of the game design software Unity, before calling Unity's mesh renderer to display the surface. VR is implemented using online assets provided by Steam VR.

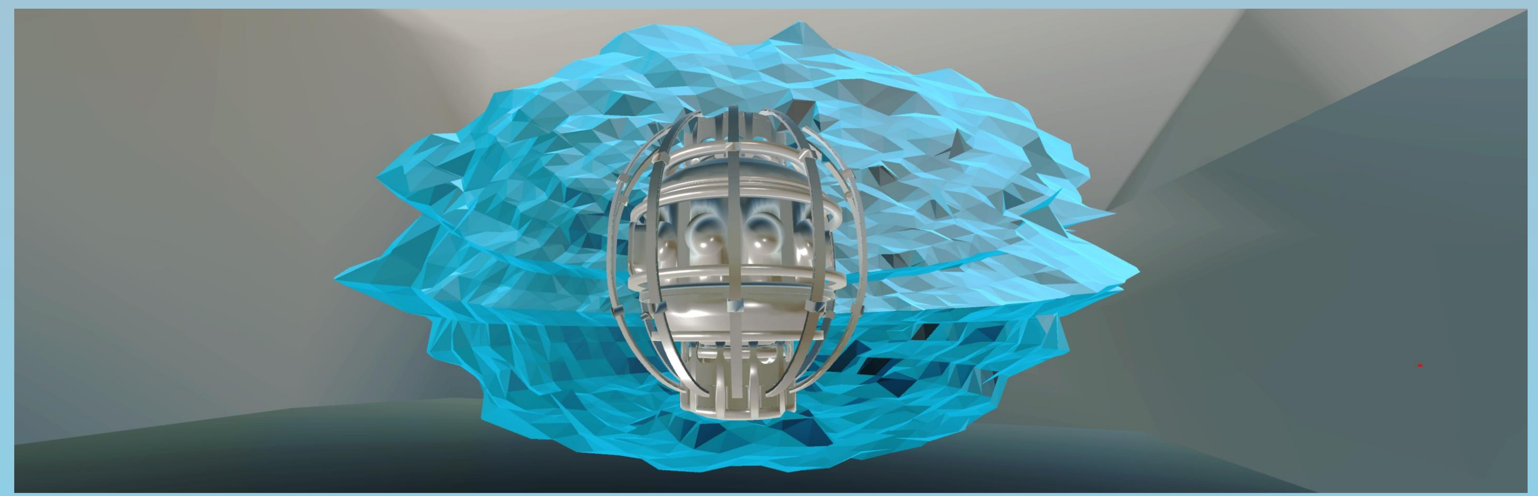

Fig. 6 | Unity Visualization of ^{28}Si Resonance Neutron Flux. The isosurface for resonance neutrons at 6.3714e10 neutrons · cm^{-2} · s^{-1} is shown and matches the outline for the surface seen in figure 2 (b).

VR Flux Assessment // Conclusions

VR inspection of isosurfaces shows well defined flux. The algorithm currently processes 229,910 grid points in ~1.5 s. Larger data sets of ~1.8 million take ~11 s to render and require performance optimization to maintain smooth VR movement.

References // Acknowledgements

1. Bourke, Paul. "Polygonising a Scalar Field." Paulbourke.net, 1994, http://paulbourke.net/geometry/polygonise/
2. Jmtrivial. "MarhcingCubes.svg". Digital Image. Wikipedia.org. Web. 15 October 2006. https://en.wikipedia.org/wiki/Marching_cubes#/media/File:MarchingCubes.svg.
3. "Attila User's Manual". Varex Imaging. PDF. 28 September 2018.
4. Nuclear Energy Agency. JANIS [database]. Retrieved from http://www.oecd-nea.org/janisweb/search/endf?inc=N&lib=ENDF%2FB-VIII.0.

Special thanks to Jonathan Klabacha, Scott Doskoczynski, and the PPPL science ed department.

Contact: JamalJ800@gmail.com, Jklabach@pppl.gov