Flux and Geometry Drivers

Joshua Isaacson with Luke Pickering

Workshop on Neutrino Event Generators

16 March 2023

Overview

- Neutrino experiments are unique in terms of the interdependency of the flux and detector geometry
- Most neutrinos pass through the detector without interacting, which in turn adds complications in ensuring weights are correctly calculated
- This talk will focus on the problem of geometry
- Goal is to develop an open source geometry tool that any generator (or lightweight theory calculation) can use to correctly place interactions in a geometry
- Questions related to handling the flux are left to the discussion

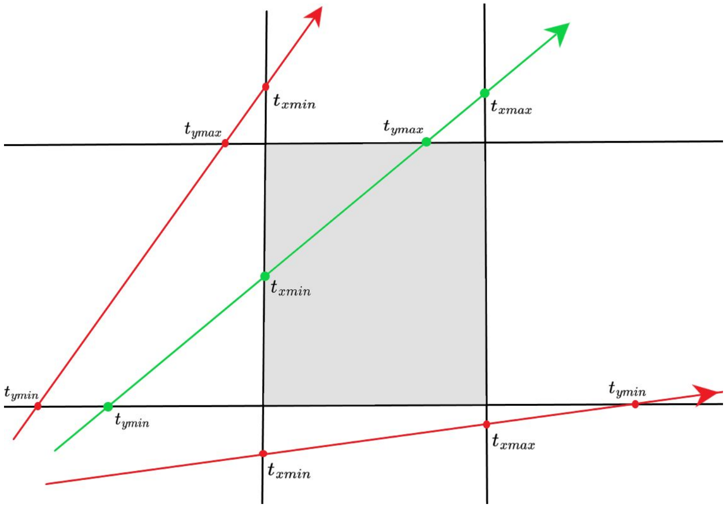
Defining a Geometry

- To be friendly to theorists, avoid dependency on ROOT and GEANT4
- GDML format is perfect for describing all components of the geometry
- Implement a GDML parser:
  - The implemented parser currently does not validate that the GDML is properly structured (ROOT and GEANT4 exist for this)
  - Can currently parse all needed structures to define:
    - Constants
    - Positions
    - Rotations
    - Materials
    - Solids
    - Volumes (except for constructive solid geometries)
    - Physical Volumes
  - Some work still left to ensure that CSGs are handled correctly

```xml
<define>
  <position name="CScint_1inToppos" x="0" y="0" z="0" unit="cm"/>
  <position name="CScint_2inToppos" x="0" y="0" z="75" unit="cm"/>
  <position name="CScint_3inToppos" x="0" y="25" z="0" unit="cm"/>
  <position name="CScint_4inToppos" x="0" y="-25" z="0" unit="cm"/>
</define>
<materials>
  <element Z="6" formula="C" name="carbon">
    <atom value="12.0107"/>
  </element>
  <element Z="7" formula="N" name="nitrogen">
    <atom value="14.0671"/>
  </element>
  <element Z="8" formula="O" name="oxygen">
    <atom value="15.999"/>
  </element>
  <element Z="1" formula="H" name="hydrogen">
    <atom value="1.00794"/>
  </element>
  <element name="argon" formula="Ar" Z="18">
      <atom value="39.9480"/>
  </element>

  <material formula="" name="CScint">
    <D value="1.043"/>
    <composite n="0.922" ref="carbon"/>
    <composite n="0.076" ref="hydrogen"/>
    <composite n="0.0006" ref="nitrogen"/>
    <composite n="0.0007" ref="oxygen"/>
  </material>

  <material formula="" name="Air">
    <D value="0.001225"/>
    <fraction n="0.781154" ref="nitrogen"/>
    <fraction n="0.209476" ref="oxygen"/>
    <fraction n="0.00934" ref="argon"/>
  </material>

</materials>
<solids>
  <box name="Top" x="300" y="300" z="300" lunit="cm"/>
  <box name="CScint0x2" x="100" y="20" z="50" lunit="cm"/>
  <box name="CScint0x4" x="100" y="20" z="10" lunit="cm"/>
</solids>
<structure>
  <volume name="CScint0x1">
    <auxiliary auxtype="SensDet" auxvalue="CScintBlob"/>
    <materialref ref="CScint"/>
    <solidref ref="CScint0x2"/>
  </volume>
```

Raytracing (https://raytracing.github.io/)

- Neutrinos are propagated through the detector using raytracing techniques.
- A set of line segments for all materials the neutrino passes through are collected
- The mean free path of the neutrino through the detector is calculated using:
  - The cross section for each material:
    - Takes into account all elements and their respective mass fractions
    - Requires the energy of the neutrino
  - The density of the given material
- The total interaction probability is obtained by summing the interaction probability in each material
- TODO: Ensure that we can get weight rescaling correct and that it meshes with input flux exposure descriptions correctly

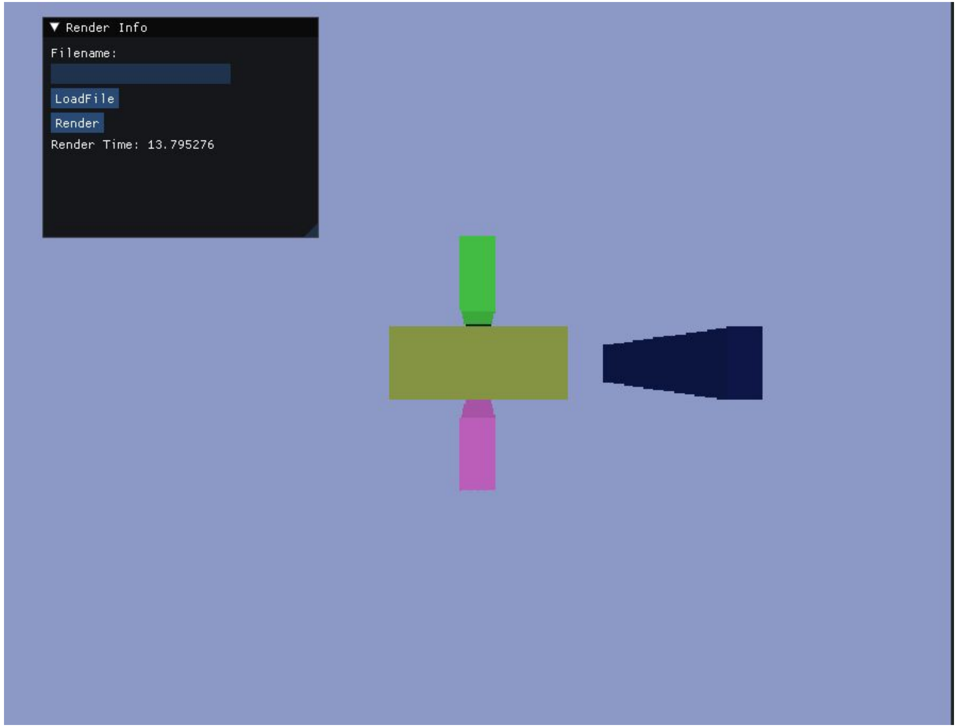


Interfacing with Flux and Generators

- Flux:
  - Ray tracer asks for a neutrino position and four-momentum
  - Ray tracer reweights POT based on probability to interact and increments it
- Event Generator:
  - Returns total cross section on each element in the detector for a given neutrino energy
  - Detector determines interaction location and element
  - Event generator creates an event with the given incoming neutrino energy

Raytracing (Validation)

- Implement visual raytracing to validate parsing is handled correctly
- Start with simple geometry
- Work in progress:
  - Draw internals of volumes
  - Draw paths neutrinos take from a given flux



Fermilab

Open Questions for the Community

- Is the community supportive of a minimal community maintained geometry driver?
- What features are needed / wanted for such a tool?
- Are theorists interested in having a simple interface to hook-up to in order to test out their models in more realistic scenarios?
- Is there support from the neutrino community to help develop a common beam simulation format along side the beam simulation community?

Conclusions

- Start of a geometry driver code
- Can construct arbitrary geometries from GDML files without ROOT or GEANT4
- Ability to place interaction vertices is on-going work
- Ability to visually inspect that the detector looks as expected is available
- Need to develop ability to handle different flux formats