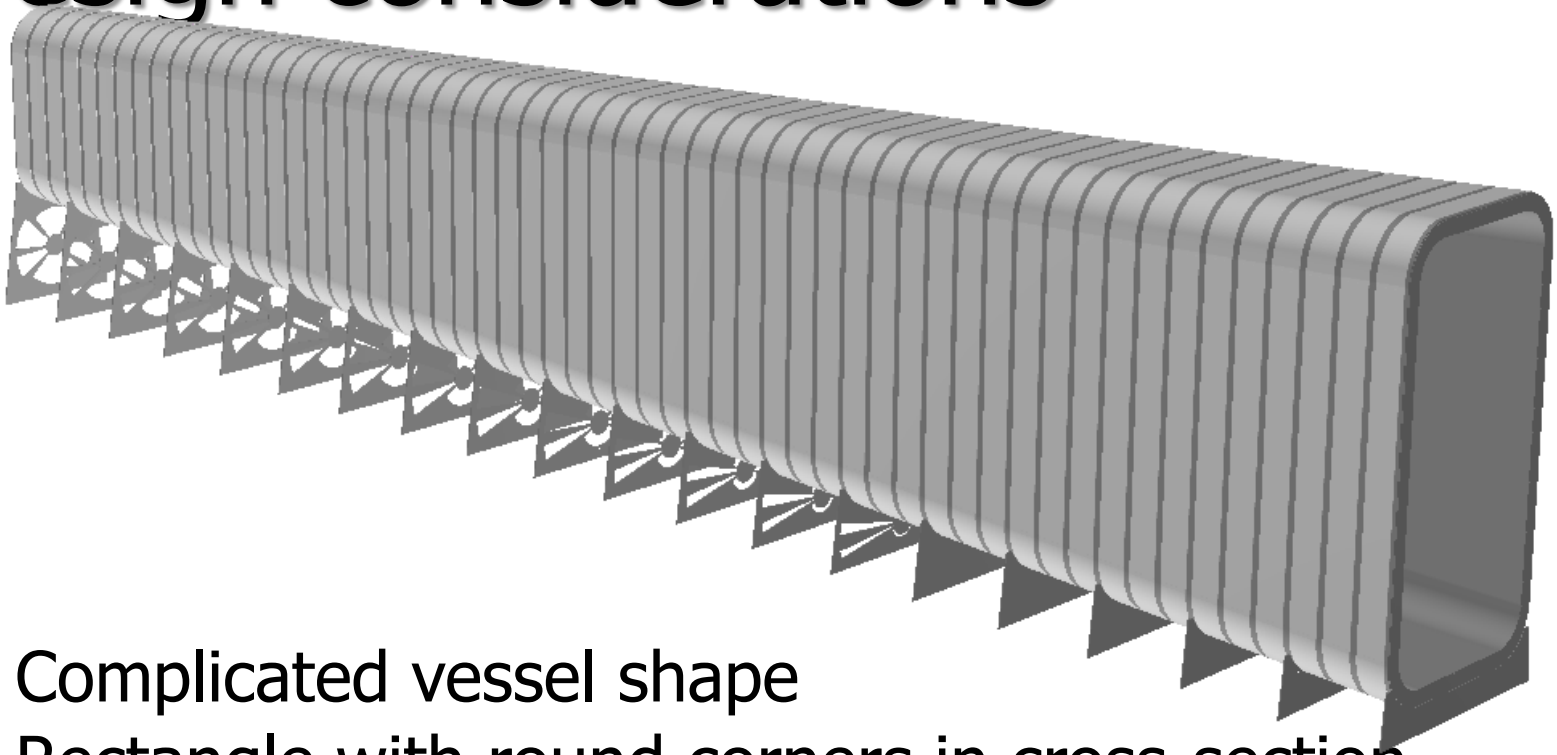
Plastic scintillator implementation

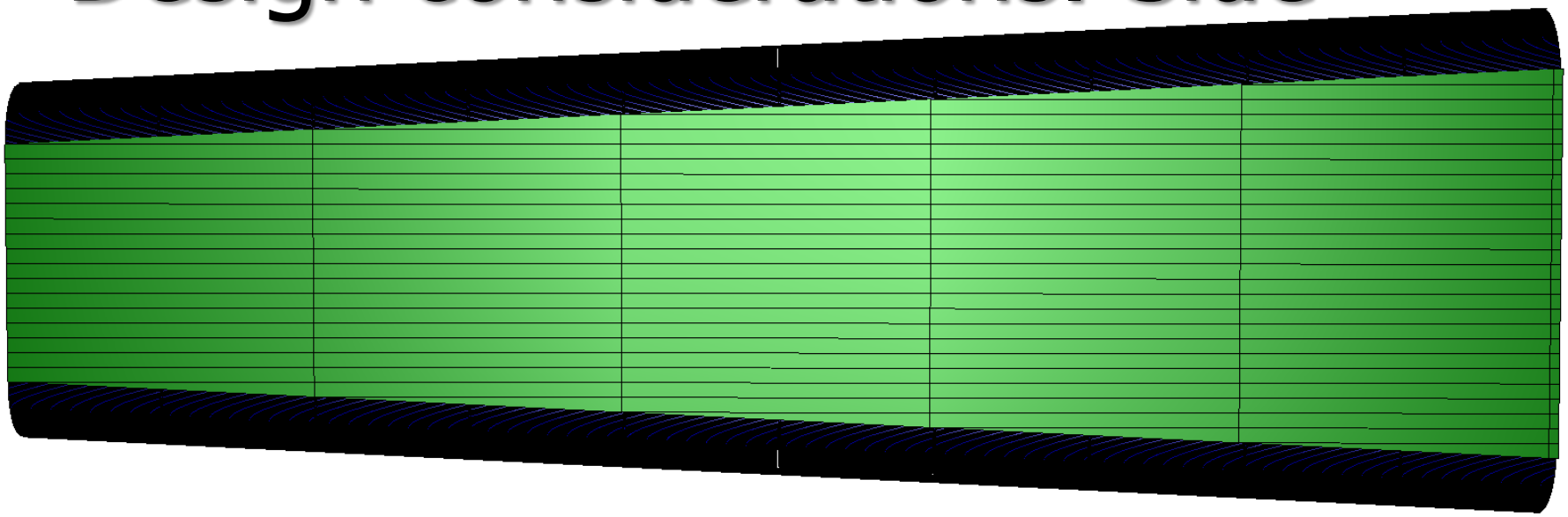
Misha Prokudin

Design considerations



- Complicated vessel shape
- Rectangle with round corners in cross-section
  - not exactly
- Arb8 shape with subtracted quarters (-2° to 92°) of tubes at the corners
  - a lot of Boolean operations will slow down the simulation
- Develop something which can be constructed with scintillator strips

Design considerations. Side

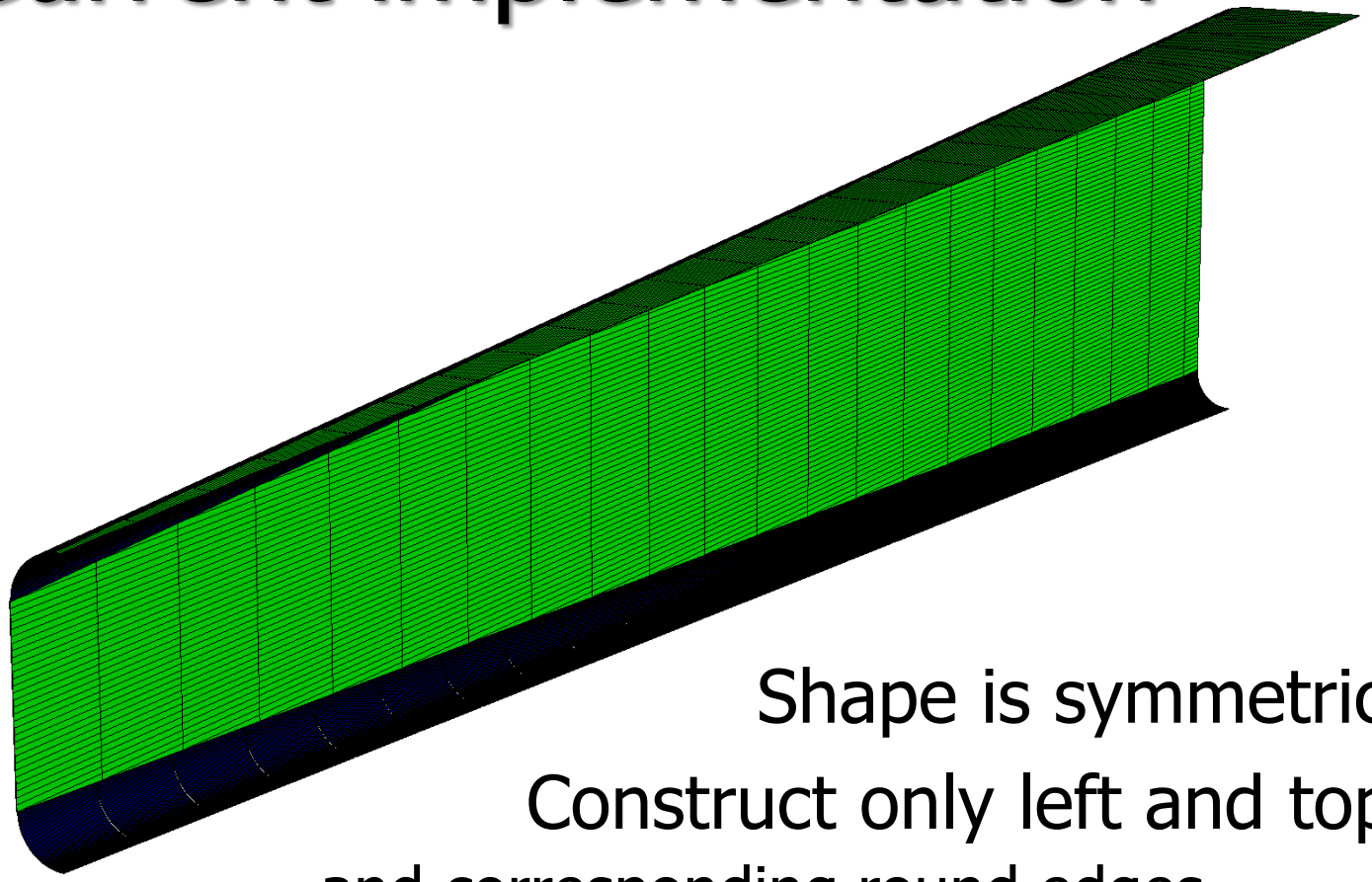


- Scintillator tiles: rectangular and triangle
- Use 3 shapes in modeling: rectangular, triangle up and down
- Integer number of tiles at the beginning and end of shape. Integer number of tiles along the length in ideal case.
  - Slope is given, fixing the number of tiles at the beginning fixes size of the tile.
  - Cut the shapes at the end on the vessel.

Design considerations.

- Integer number of tiles at the beginning and end of shape. Integer number of tiles along the length.
  - Slope is given. Fixing the number of tiles at the beginning fixes size of the tile.
  - Cut the shapes at the end on the vessel.
    - unavoidable without fine tuning of veto size.
  - Different tiles at different edges of the vessel.
    - fine tuning of veto size.
  - Keep the size of tiles physical
    - Length 1-2 m and width 3-20 cm
    - Width up to 20 cm.
- Round corners: use tube segments interested with box for simplicity
  - No shape restrictions but physical sizes.
- Thickness: 1 cm

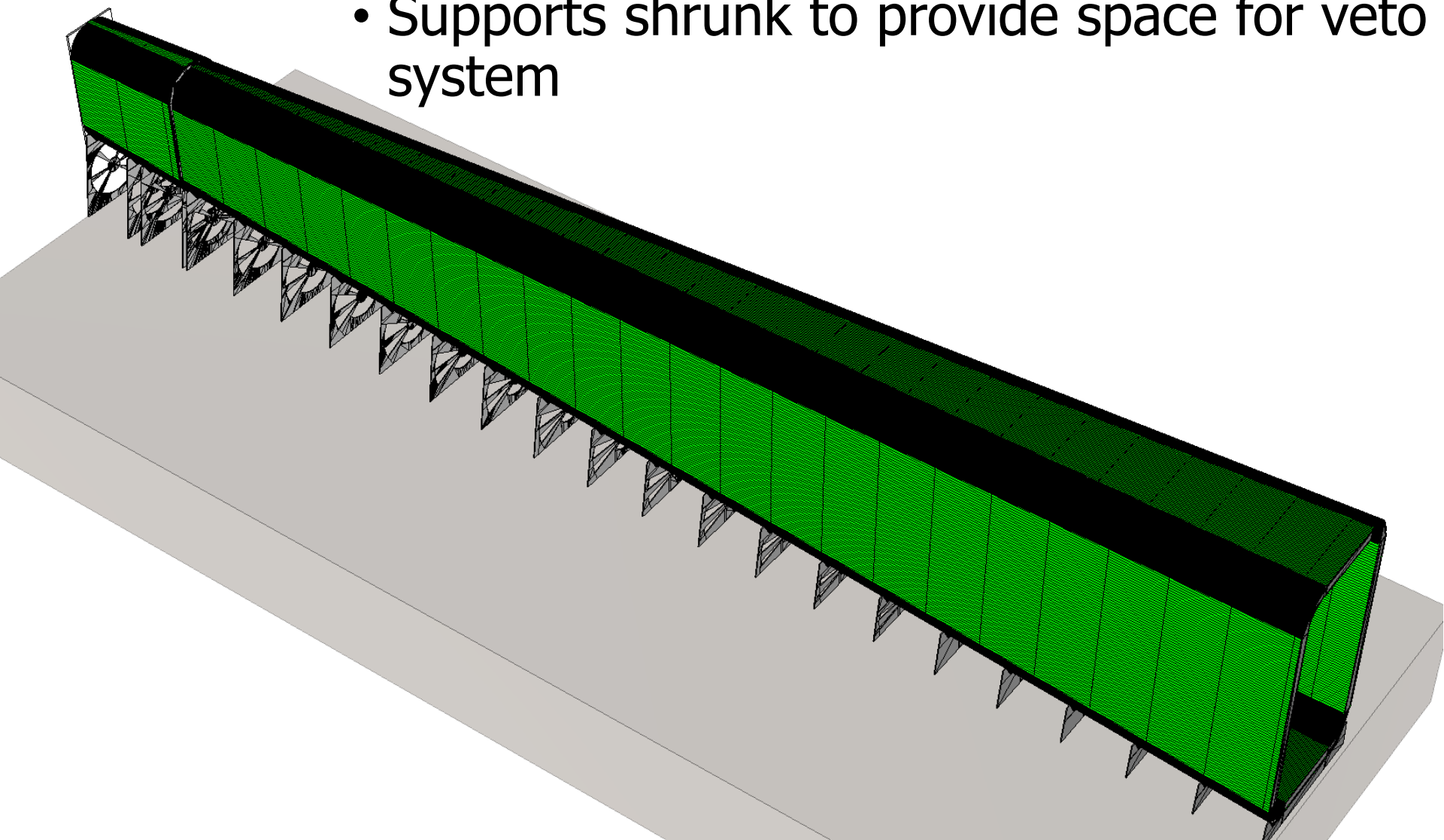
Current implementation



- Shape is symmetric
- Construct only left and top
  - … and corresponding round edges
- Two copies left_1 and left_5 (rotated) to get the full shape
  - ProcessHits() modified to handle this
    - only modification

Current implementation

- Supports shrunk to provide space for veto system



One more thing… Storage of MCTracks

- Previously: store MCTracks with E_{kin}>threshold or all MCTracks
  - Lost low energy tracks which produced an MCPoint
  - Huge disk space consumption
- Investigation of storage of MCTracks which produced at least one MCPoint
- Currently: several ways of MCTracks storage
    - implementation suggested; final solution by Thomas
  - E_{kin}>threshold
    - 151M 10k default events
  - $N_{MCPoints}$>0
    - 259M 10k default events
  - E_{kin}>threshold or $N_{MCPoints}$>0
    - 300M 10k default events
  - All MCTracks
    - 1638M 10k default events

Conclusions

- The first implementation of solid plastic veto is here
- All ingredients for simulation and reconstruction is present
- Cuts and thresholds for hit formation to be tuned
- New approach to MCTracks storage has been studied

Numbering scheme. left_1

- DetectorID in vetoPoint: ABBCCCD
- A: 1 for inner veto of T2, 2 for outer veto T2
  3 for inner veto of T1, 4 for outer veto T1
  - VolumeNames: T1DecayVolVeto+, T2OutterWallVeto+ …
- D: 0 and 2 for round corners, 1 for vertical wall, 3 for horizontal wall
  - +4 for copy (left_5)
  - VolumeNames: D=0: +VetoT2, D=2: +VeroT1, D=1: VetoY, D=3: VetoX
- CCC: X or Y number of the tile
  - VolumeNames
    - D=1 and 3: +DwTr for CCC=0, +UpTr for CCC=1, +Rect for others
    - D=0 and 2: +_DV
- BB: Z number of the tile, BB=99 for chopped tiles
  - VolumeNames:
    - D=1 and 3: +1 for BB!=99, +2 for BB=99
    - D=0 and 2: ignored