Development Towards a Rapid Analysis Plume Dynamics Estimation Tool

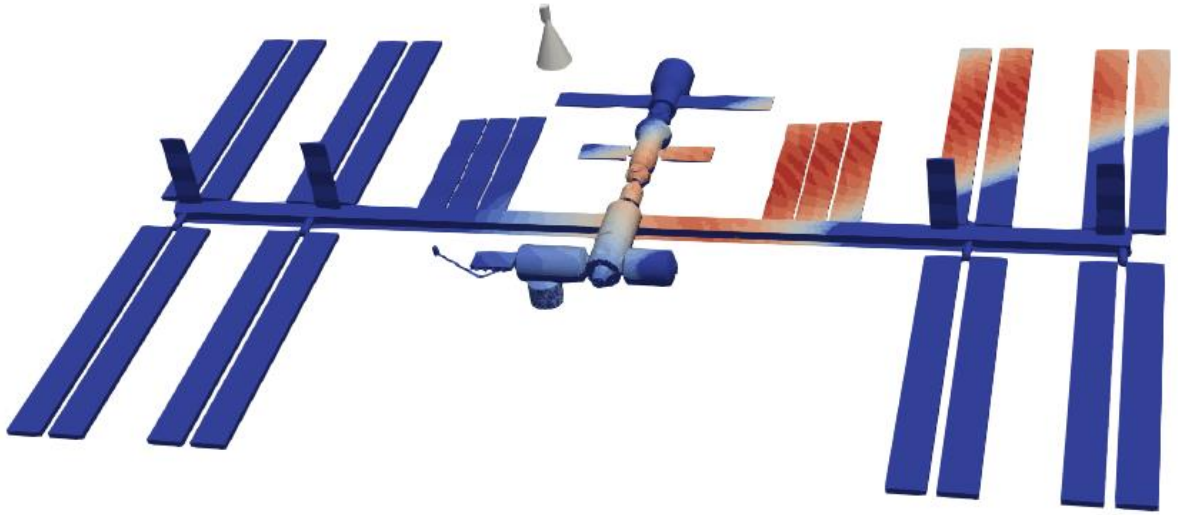
DSL Summer 2023 Intern Out-brief

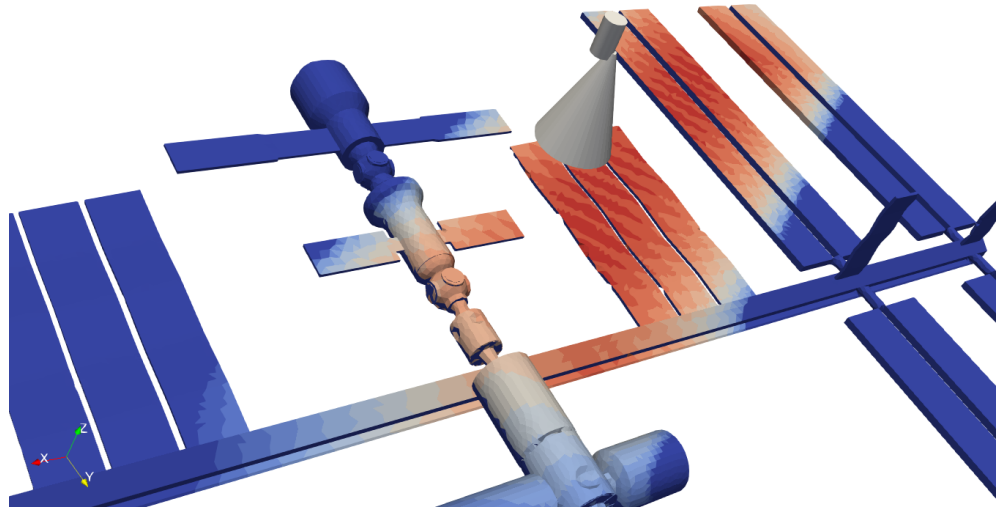
Intern: Andres "Andy" Torres-Figueroa

University of Central Florida

Senior, Aerospace Engineering B.S.

Mentor: Dr. Jonathan S. Pitt

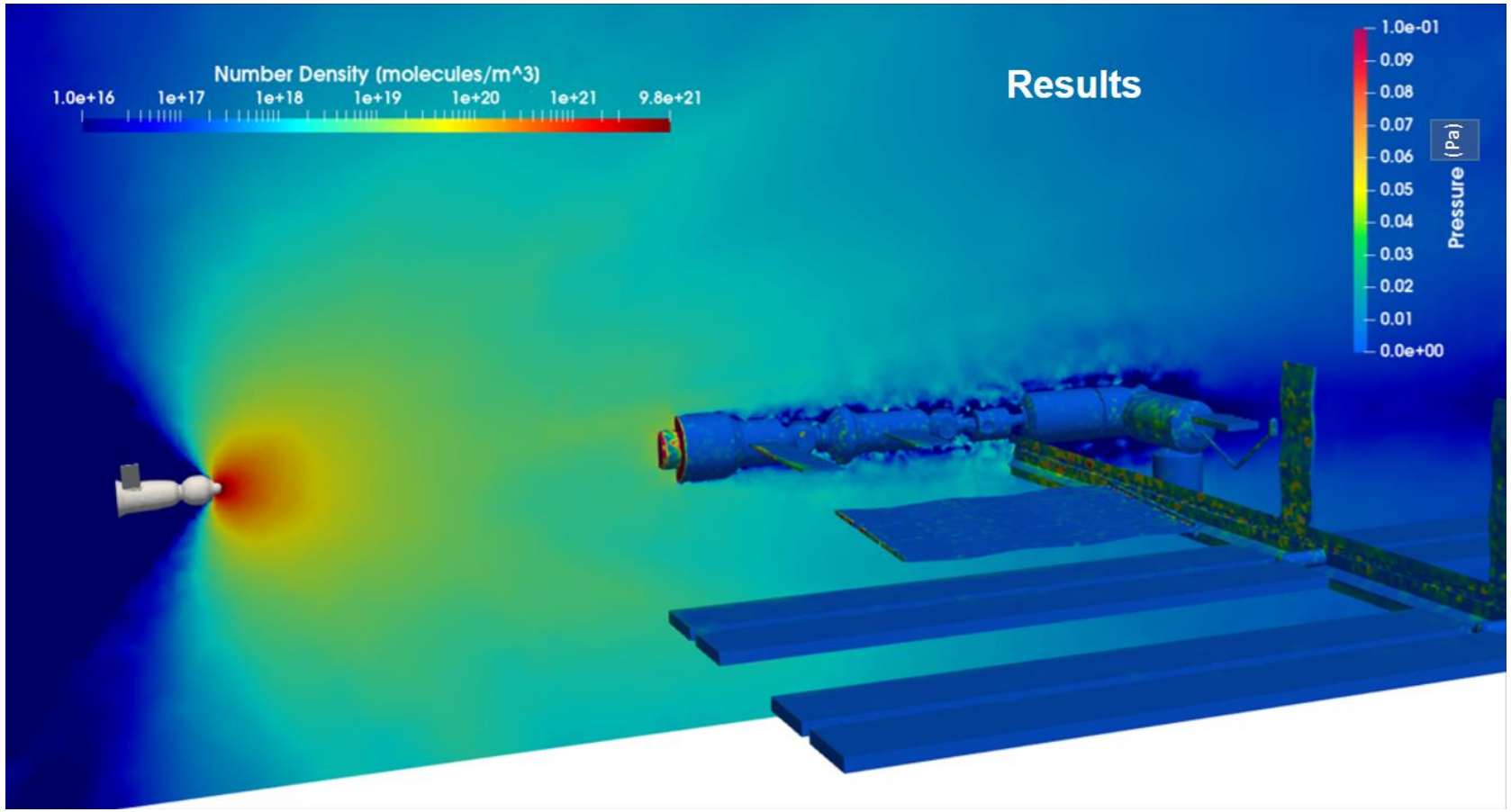




Outline

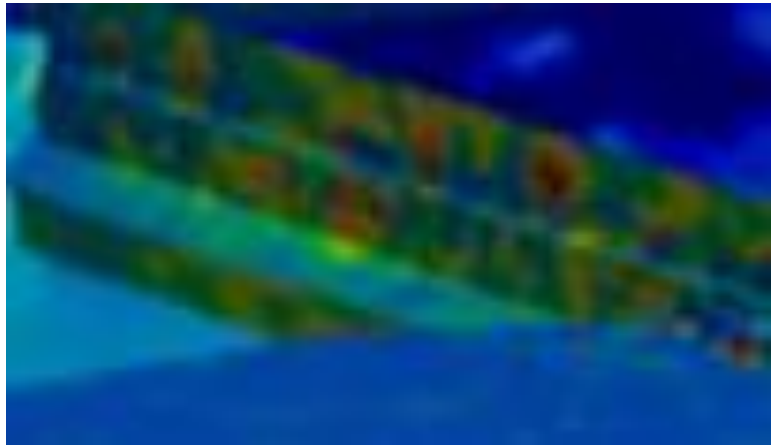
- **Background**
  - Previous Work and Motivation
- **Requirements**
  - Reading Inputs
  - Vehicle and Thruster Transformations
  - Plume Strike Logic
- **Demonstration Cases**
- **Future Steps**
- **Acknowledgements**

Previous Work and Motivation



- Spring/Summer 2022 ran high fideltiy Direct Simulation Monte Carlo (DSMC) calculations of rareified plumes.
- Gives important and detailed impingement information but can be prohbilitvely expesenve to run.
- **Thus, tools for executing engineering calculations are necessary for rapid esimtation and iteration of vehicle and trajectory designs**
- DSMC Calculations are better suited for consolidation of results and finalizing designs.

GOAL: Develop a simulation tool that uses engineering fidelity models for rapid estimation of plume strike effects and efficient iteration of early design decisions.



Requirements

Requirements	Design Decision
Program needs to be modular in complexity on of analysis, run on an HPC environment, user and developer friendly, and be able to interface various data sources.	Python chosen as programming language.
Program needs to have embedded documentation for developer support	Pdoc is a Python module
Program needs to read in and parse text files containing Jet Firings Histories (JFH) and Thruster Configuration Data (TCD) relevant to the RPOD scenario.	Python can handle using standard libraries for string manipulation.
Program needs to read in surface mesh data for a target and visiting vehicle to define an RPOD scenario, while also tranformating the STL data according to the TCD and JFH data.	numpy-stl is a library within Python that is used.
Program needs to write resulting data to files in a VTK format for easy visualization of results.	Export VTK is another library within Python is used
Program needs to allow user to configure plume physics, thruster operating conditions, mesh data, and meta data for and RPOD scenario.	Python has configuration objects, and uses Object Oriented Programming

Reading Inputs

Visiting Vehicle Data

- STL Surface Mesh
- Thruster Configuration Data
- Thruster Plume Characterization
  - Plume Physics
  - Thruster Data

Target Vehicle Data

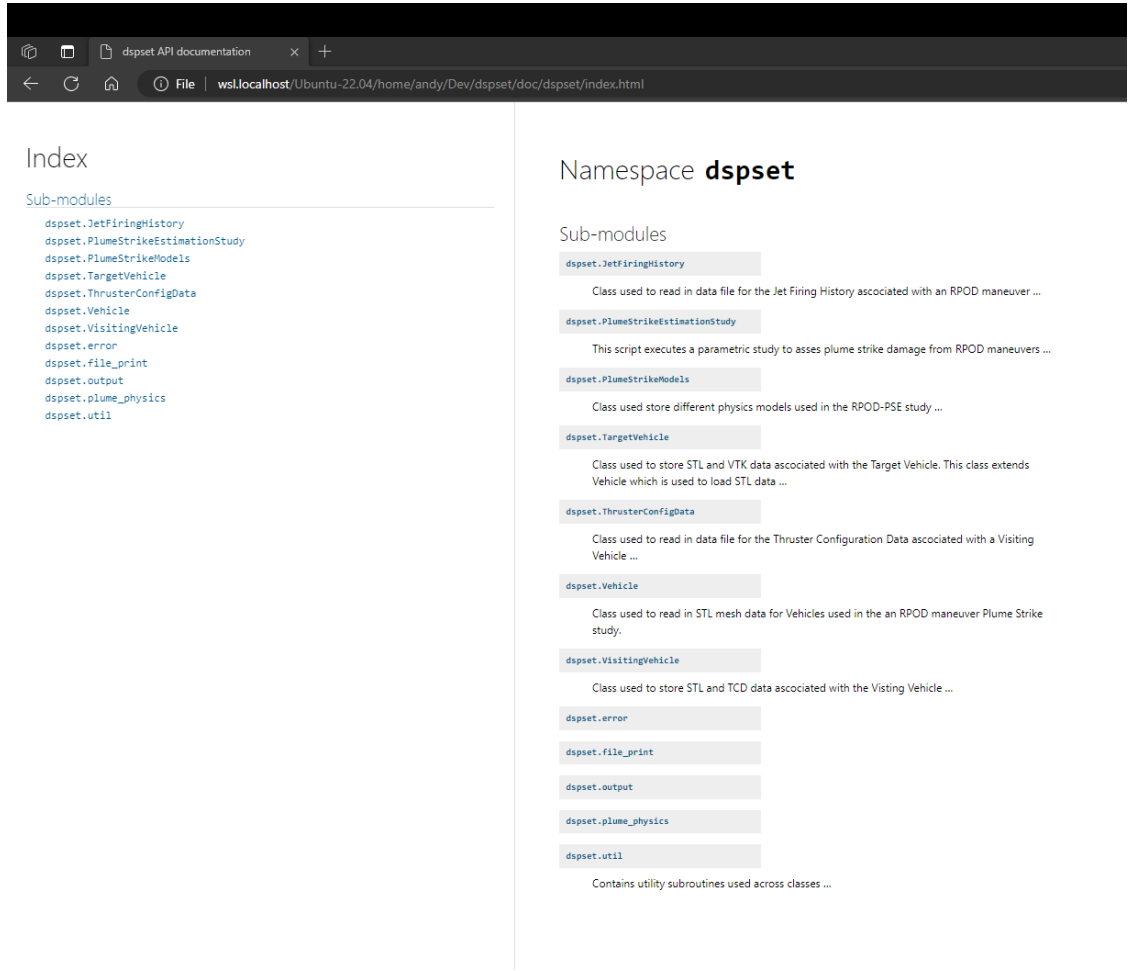
- STL Surface Mesh
- Orientation Adjustments
- Surface Information
  - Physical Properties
  - Component Grouping
  - Temperature

Plume Strike Estimation Study

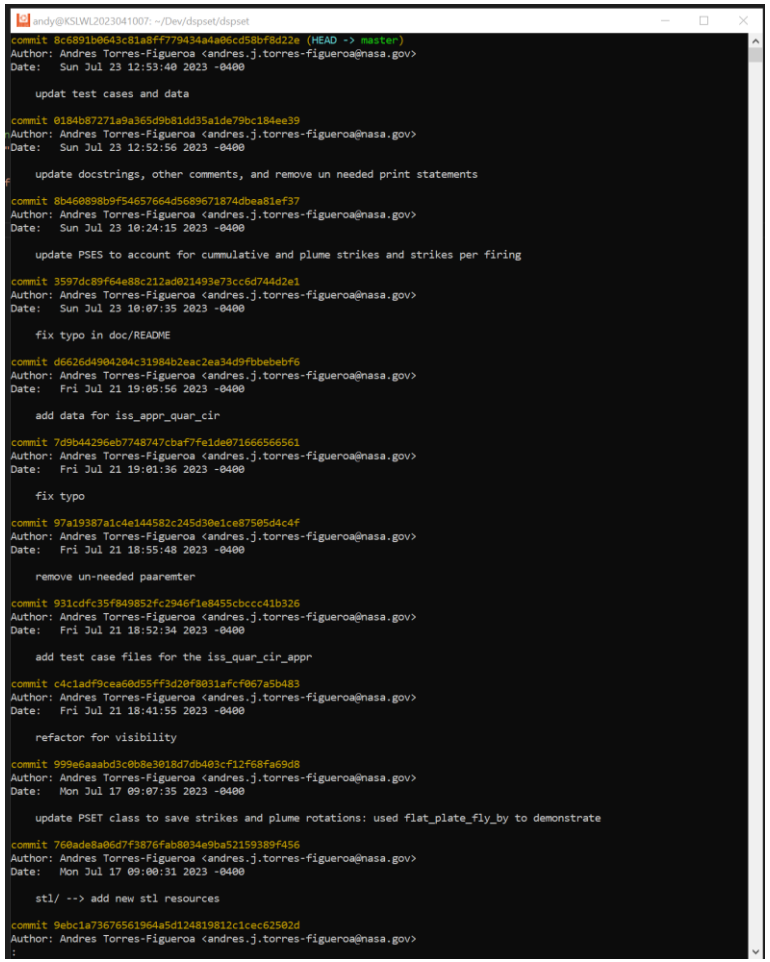
- Jet Firing History
- Main Configuration File
  - Simulation Parameters
  - Data-Post Processing Routines
  - Input/Output File Locations

Documentation

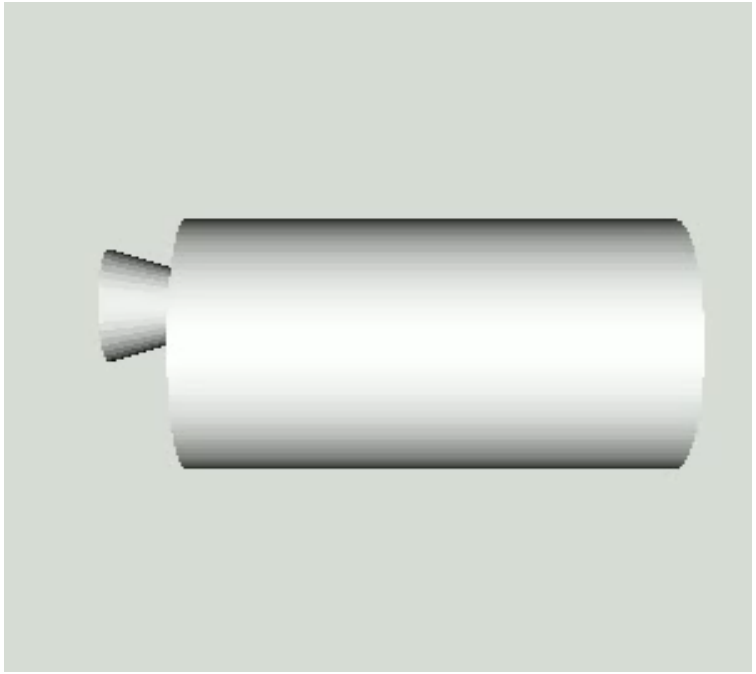
Automated Documentation Generation

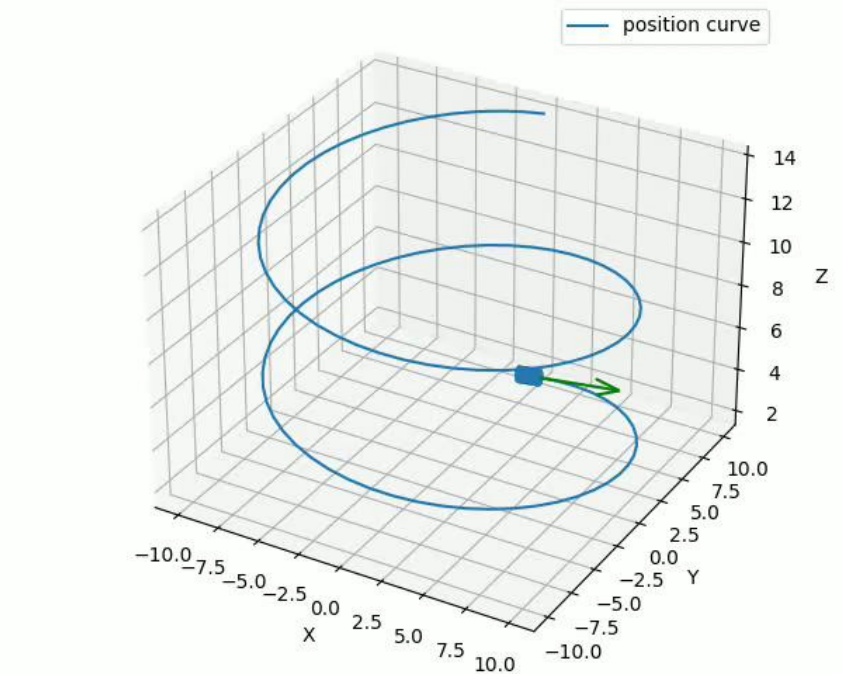


Version Control



Vehicle and Thruster Tranformations





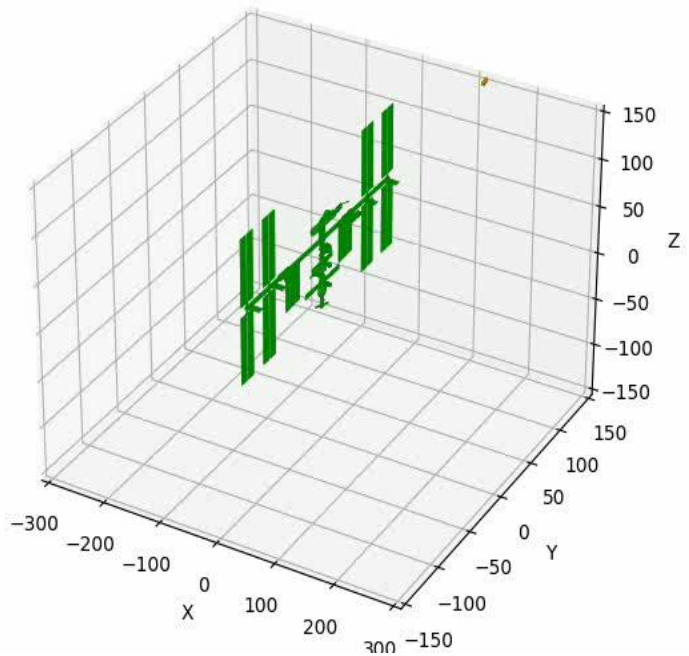
$$\begin{bmatrix} \widehat{x_{VV}} \\ \widehat{y_{VV}} \\ \widehat{z_{VV}} \end{bmatrix} = \begin{bmatrix} VV_{11} & VV_{21} & VV_{31} \\ VV_{12} & VV_{22} & VV_{32} \\ VV_{13} & VV_{23} & VV_{33} \end{bmatrix}_{VV} \begin{bmatrix} \widehat{x_i} \\ \widehat{y_i} \\ \widehat{z_i} \end{bmatrix}$$

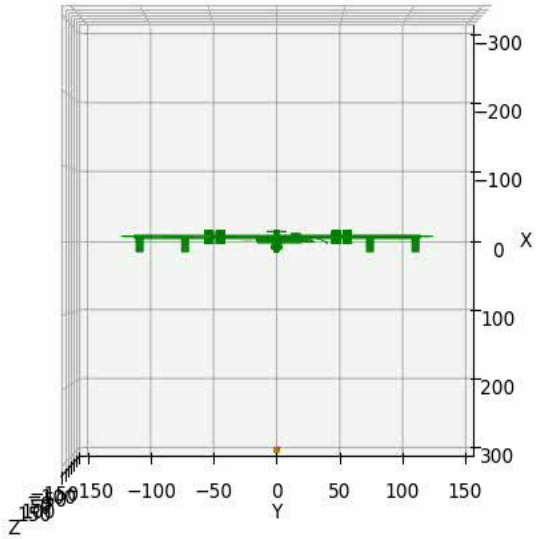
Specified For Each Thruster Using Thruster Configuration Data

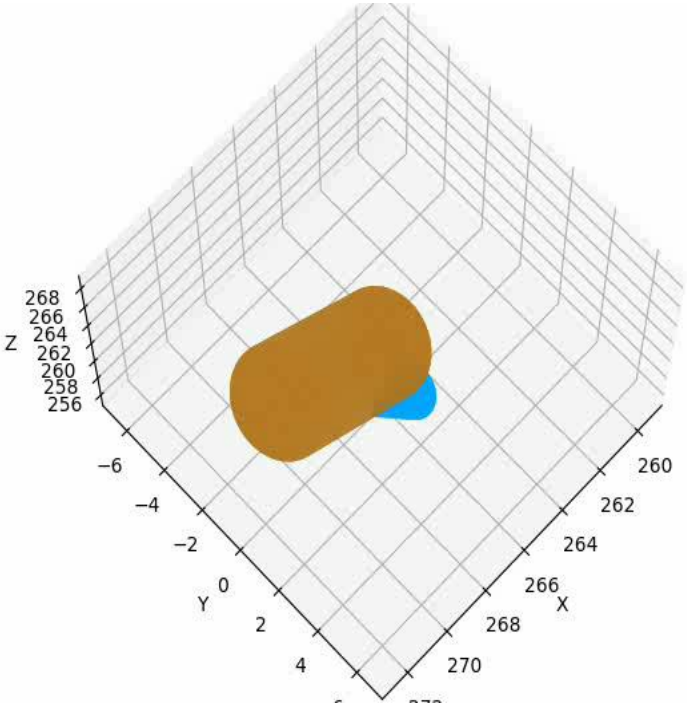
$$\begin{bmatrix} \widehat{x_{TV}} \\ \widehat{y_{TV}} \\ \widehat{z_{TV}} \end{bmatrix} = \begin{bmatrix} TV_{11} & TV_{21} & TV_{31} \\ TV_{12} & TV_{22} & TV_{32} \\ TV_{13} & TV_{23} & TV_{33} \end{bmatrix}_{TV} \begin{bmatrix} \widehat{x_{VV}} \\ \widehat{y_{VV}} \\ \widehat{z_{VV}} \end{bmatrix}$$

Specified For Each Firing Using Jet Firing History

Vehicle and Thruster Tranformations



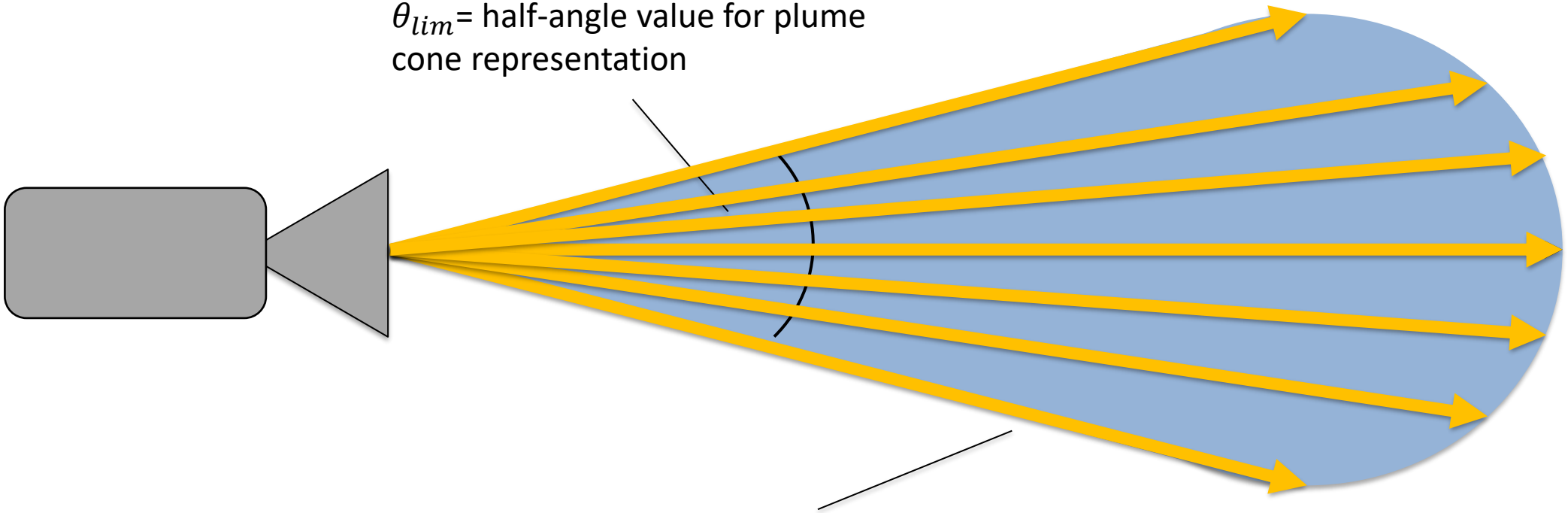




Plume Cone Configuration

1. Simple plume representation, equivalent to line-of-sight analysis.
2. Can be visualized as a sector of a sphere
3. Plume configuration variables, θ_{lim} and $d_{critical}$ are user defined parameters to establish plume cone region.

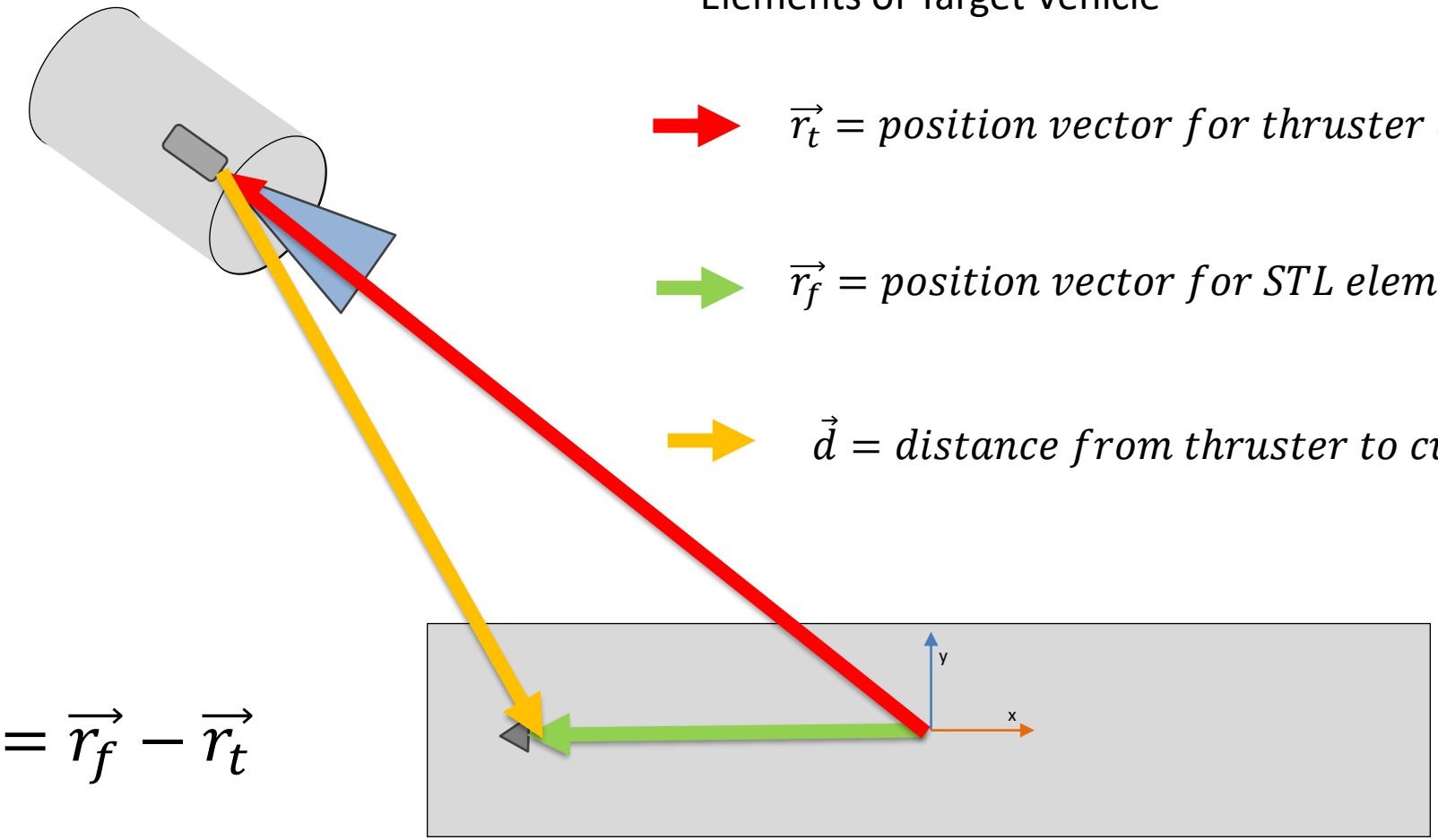
θ_{lim}= half-angle value for plume cone representation



$d_{critical}$= max radial distance of plume

Plume Strike Logic

Distance Vector from RCS Thruster to Surface Elements of Target Vehicle



$\vec{r_t} = position\ vector\ for\ thruster\ exit\ wrt\ to\ TV$

$\vec{r_f} = position\ vector\ for\ STL\ element/face\ centroid\ wrt\ to\ TV$

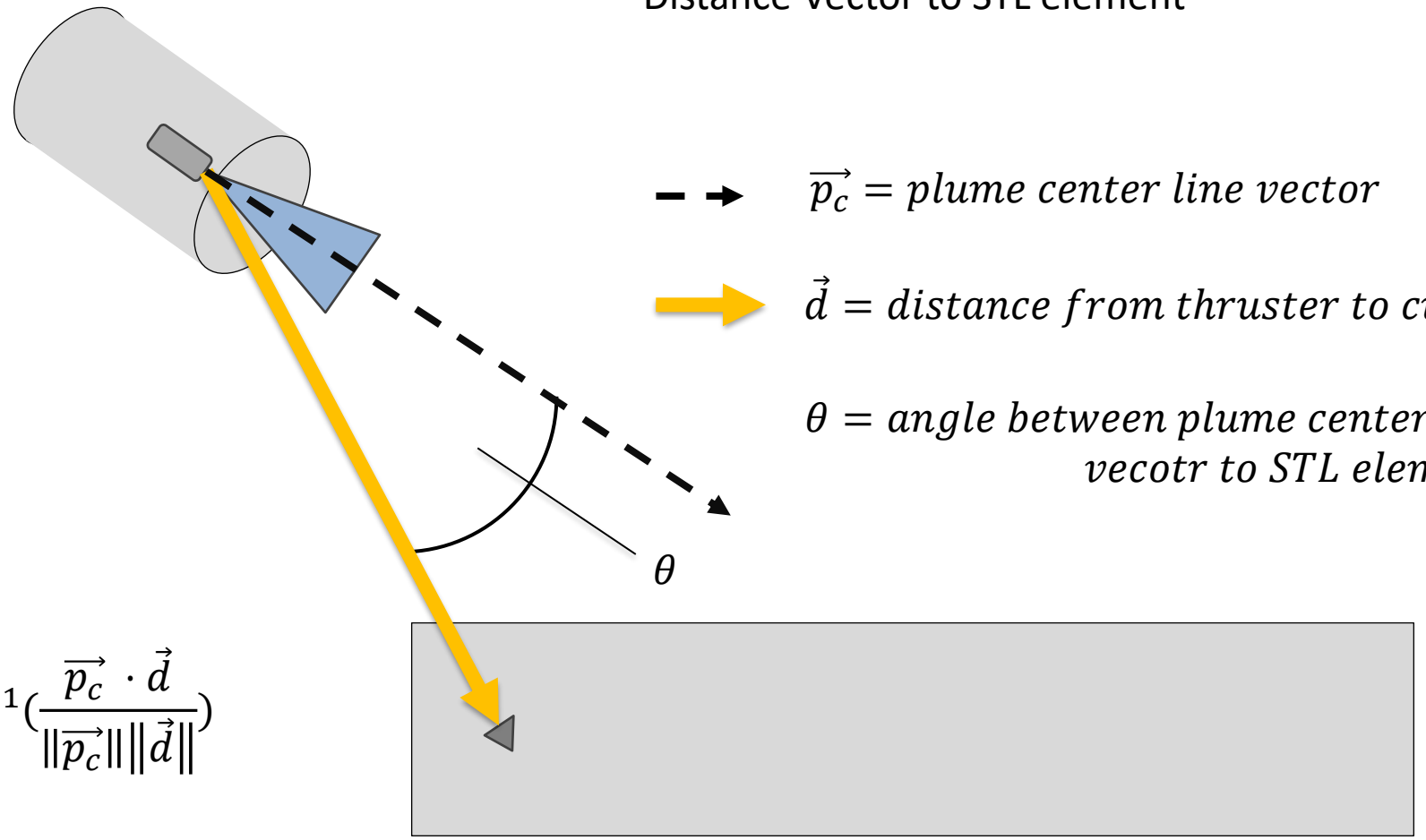
$\vec{d} = distance\ from\ thruster\ to\ current\ STL\ element\ centroid$

$$\vec{d} = \vec{r_f} - \vec{r_t}$$

$if\ ||\vec{d}|| < d_{critical}$
$STRIKE$
$else$
$NO\ STRIKE$

Plume Strike Logic

Angle Between Plume Centerline and
Distance Vector to STL element



$\overrightarrow{p_c} = plume\ center\ line\ vector$

$\vec{d} = distance\ from\ thruster\ to\ current\ STL\ element\ centroid$

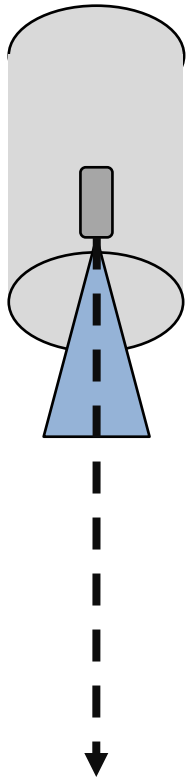
$\theta = angle\ between\ plume\ center\ line\ and\ distance\ vecotr\ to\ STL\ element$

$$\theta = \cos^{-1}\left(\frac{\overrightarrow{p_c} \cdot \vec{d}}{\|\overrightarrow{p_c}\|\|\vec{d}\|}\right)$$

$if\ \theta < \theta_{liniting}$
$STRIKE$
$else$
$NO\ STRIKE$

Plume Strike Logic

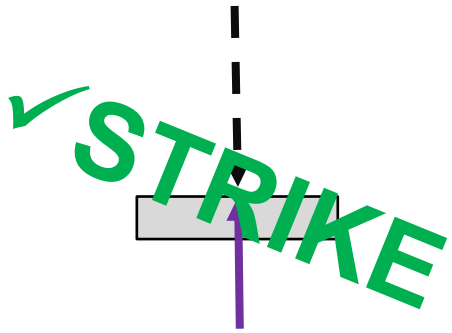
Face Orientation Check:
Pre-Cursor to Shading
Calculations



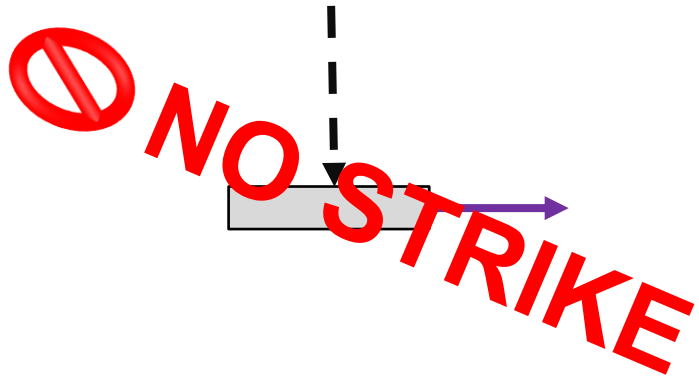
Dot Produjct
Evalutation
$\overrightarrow{p_c} \cdot \vec{n}$

$\overrightarrow{p_c} = plume\ center\ line$

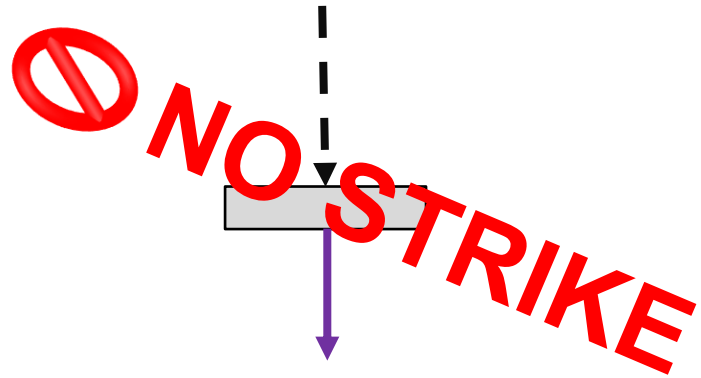
$\vec{n} = normal\ vector\ for\ STL\ element$



Facing Each Other
$\overrightarrow{p_c} \cdot \vec{n} = -1$



Orthogonal
$\overrightarrow{p_c} \cdot \vec{n} = 0$



Facing Same Direction
$\overrightarrow{p_c} \cdot \vec{n} = 1$

Plume Strike Logic

Pseudocode

For each **firing** in the **JFH**:

1. Collect **VV transformation** data
2. Determine which **thrusters** are **firing** from the JFH.

For each **active thruster** in the current **firing**:

1. Transform **thruster** according to the VV and JFH descriptions
2. Extract **position vector** for thruster exit.
3. Extract **normal vector** for **plume center line**

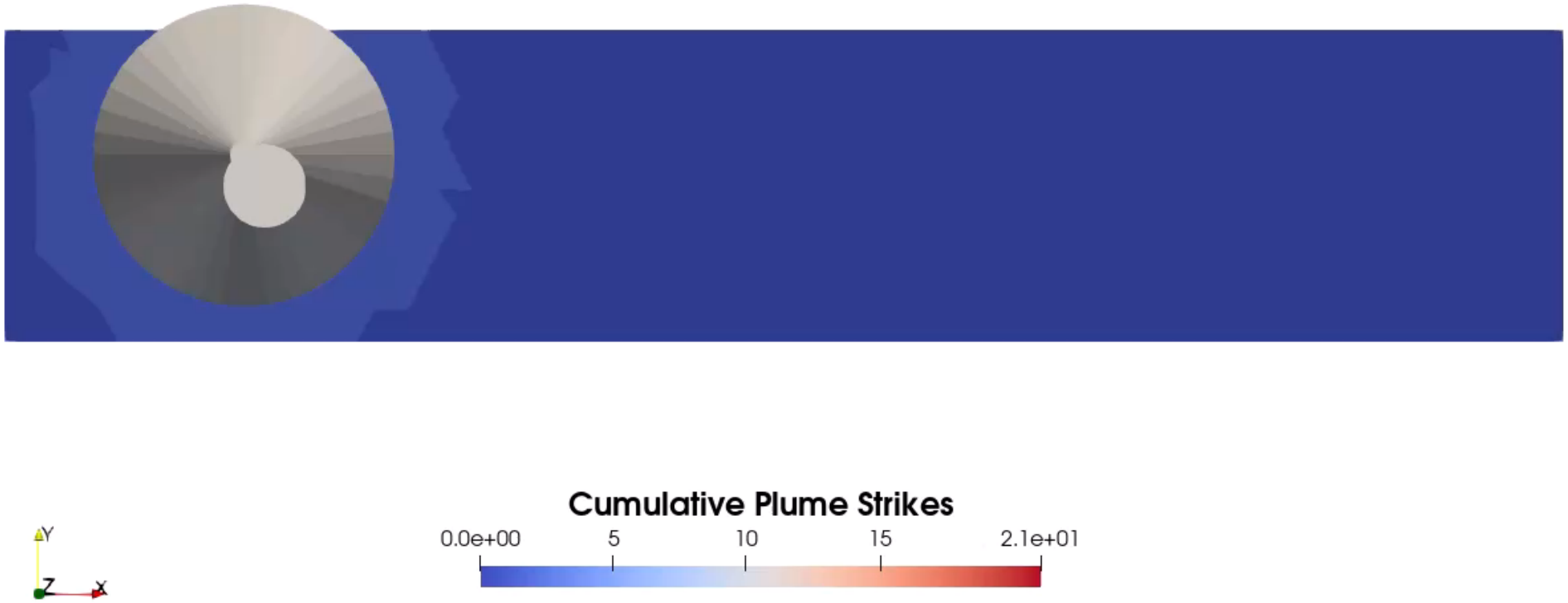
For each **face** in the **TV surface mesh**:

1. Calculate **centroid coordinate** of current face.
2. d = Calculate **distance vector** between thruster exit and centroid coordinate.
3. theta = Calculate **angle between** distance vector and the plume's center line vector.
4. dot product = Calculate **dot product** between plume center line and the normal vector of the current stl element

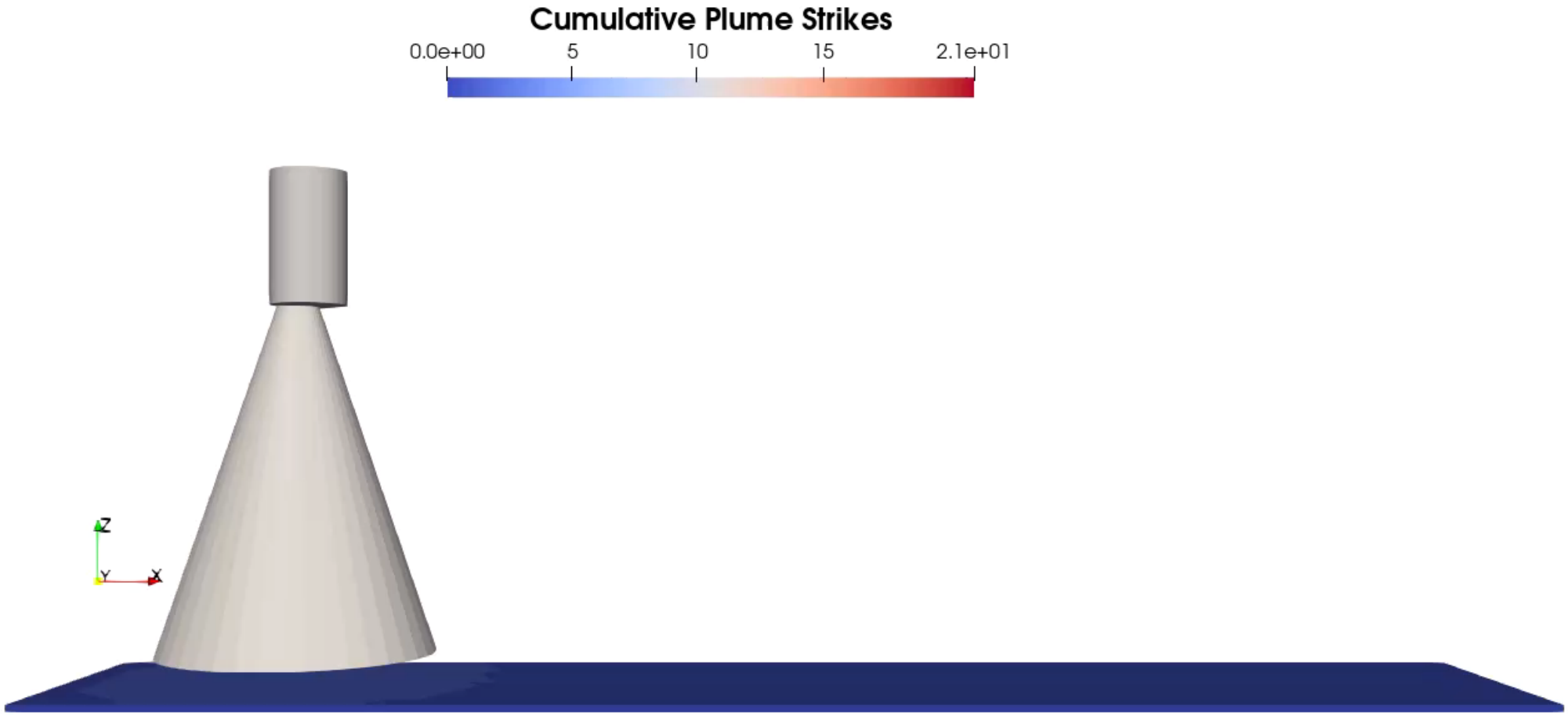
if (magnitude of d < critical distance) and (theta < limiting theta) and (dot product < 0):
    add strike to current stl element
else
    do not add a strike to the current stl element

Demonstration Case:Flat Plate Fly By

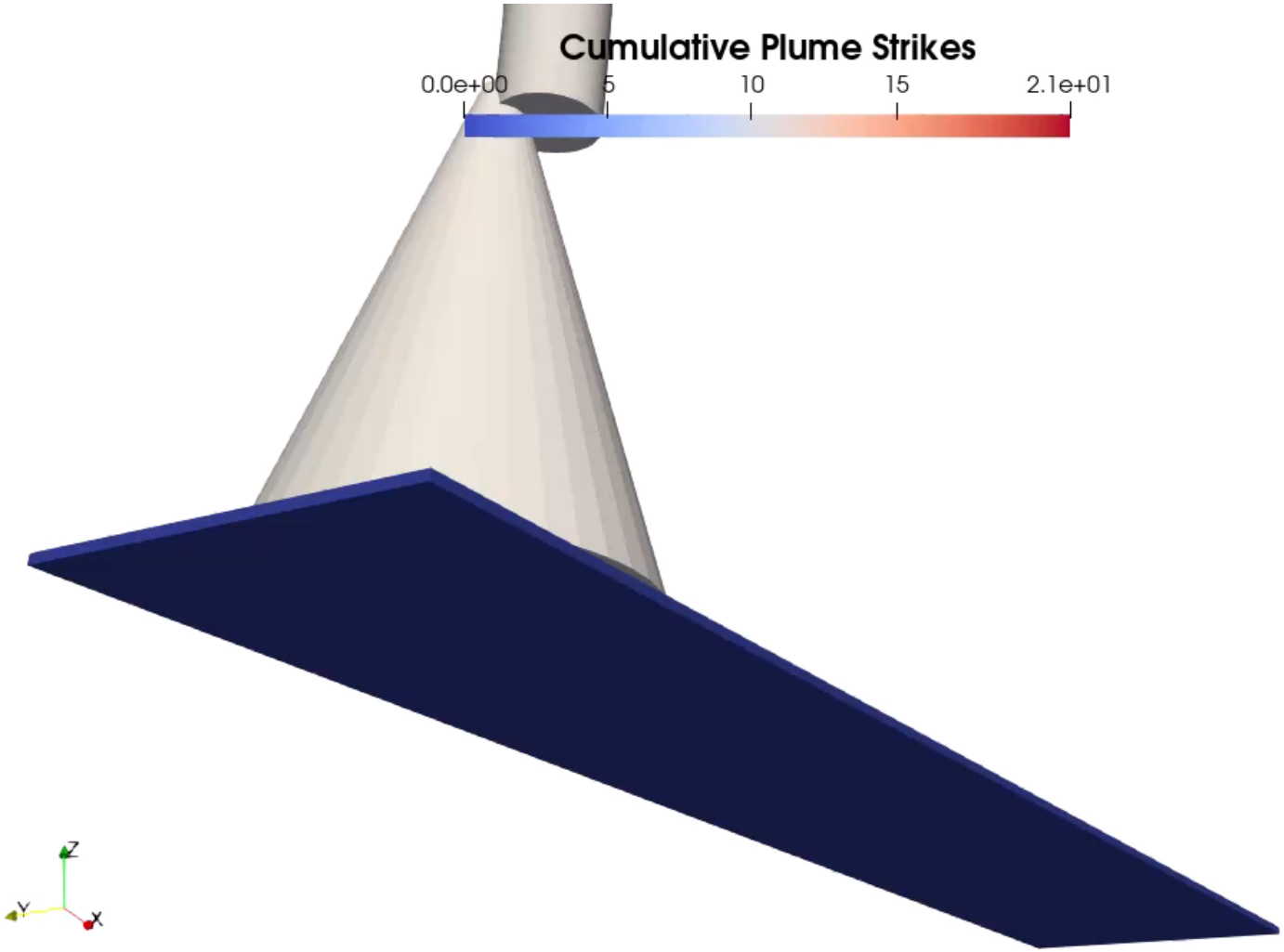
Plume Max Distance: 25 m
Plume Half-Angle: 25 degrees
Time Steps: 50
Run Time: 9 seconds



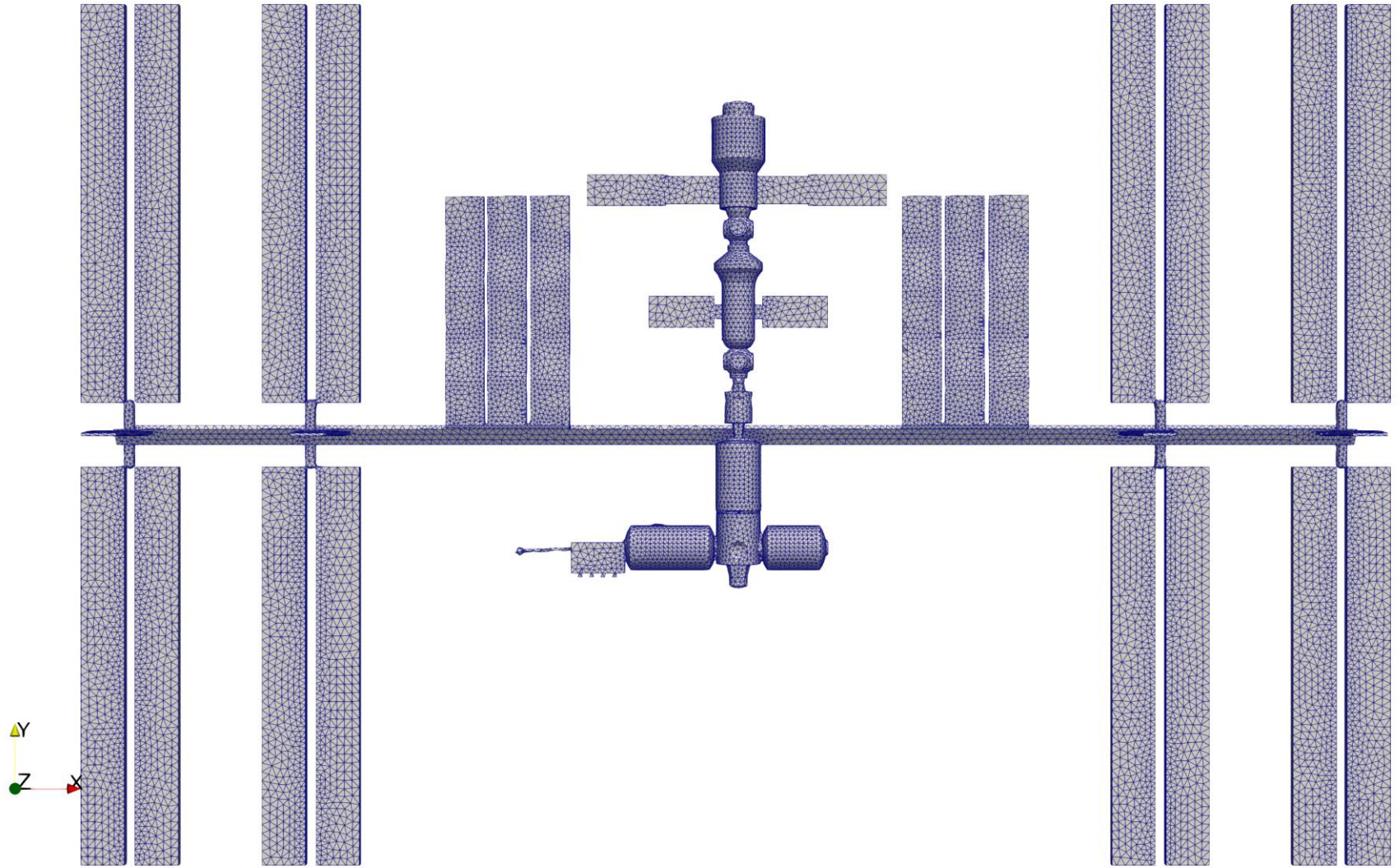
Demonstration Case: Flat Plate Fly By



Demonstration Case: Flat Plate Fly By

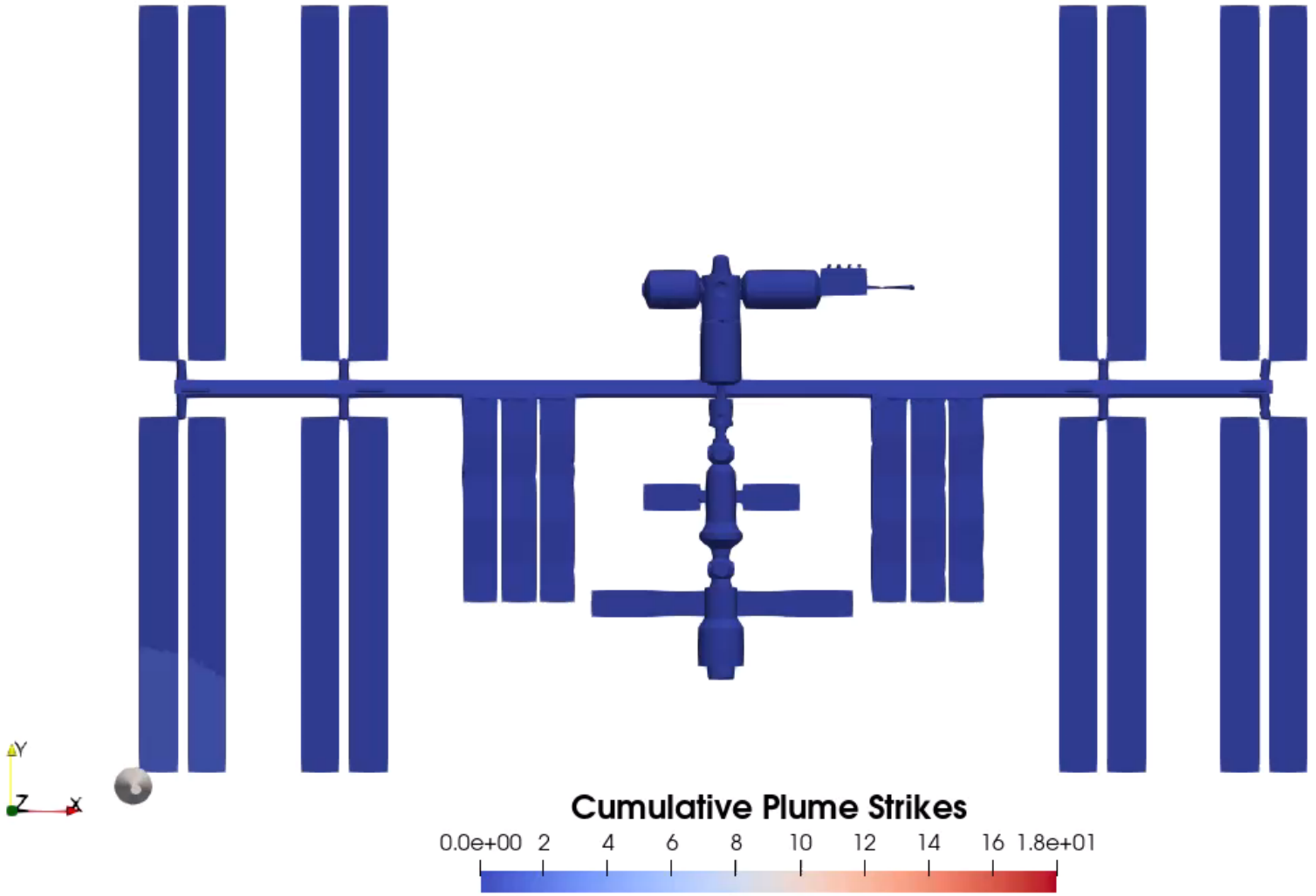


ISS STL Surface



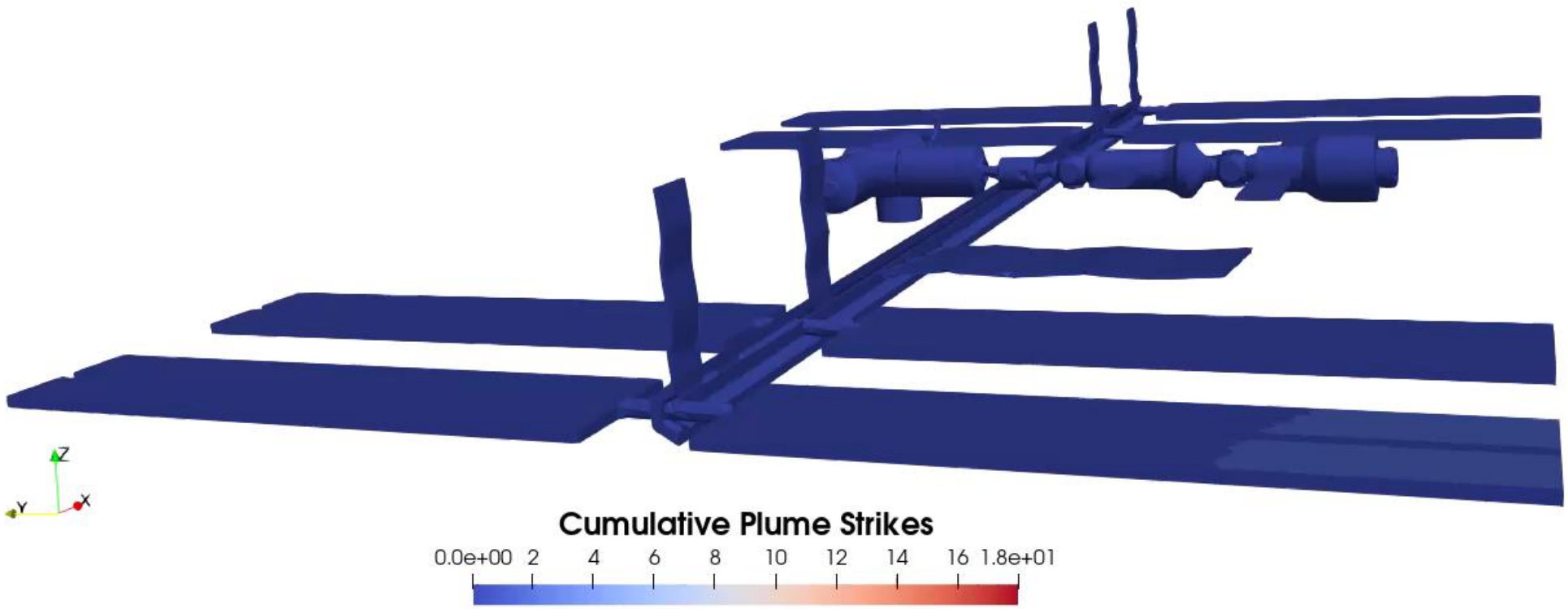
Cell Count:75,000 faces

Demonstration Case: ISS Fly By



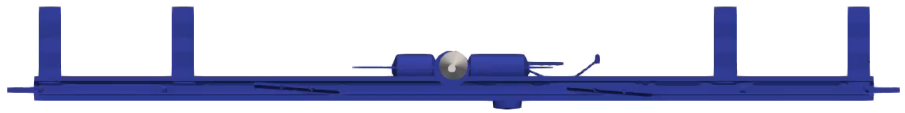
Time Steps: 100
Run Time: 5 minutes

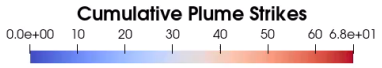
Demonstration Case: ISS Fly By

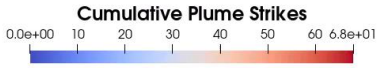


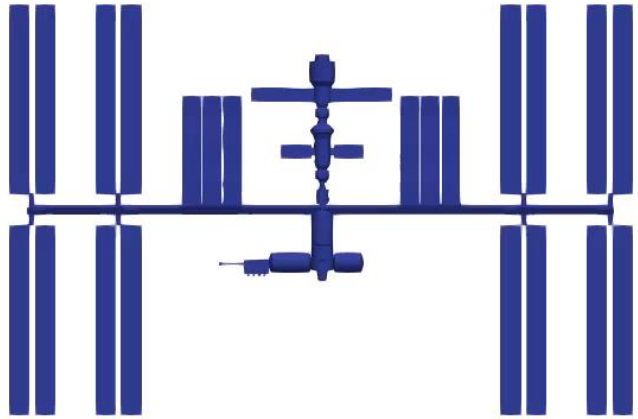
Demonstration Case: ISS Horizontal Docking

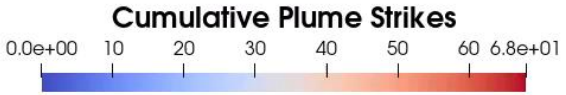
Plume Max Distance: 150 m
Plume Half-Angle: 45 degrees
Time Steps: 100
Run Time: 5 minutes



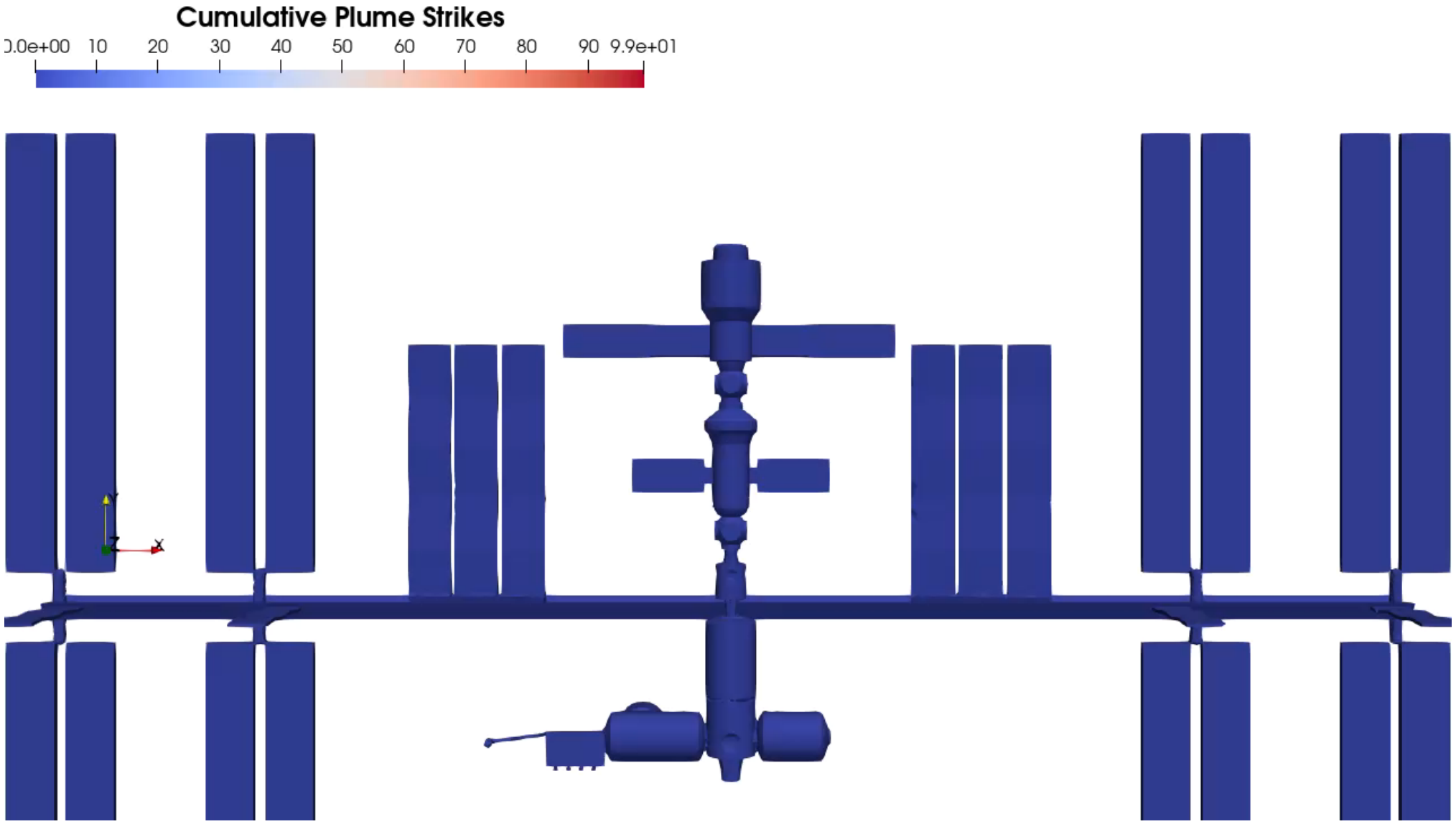




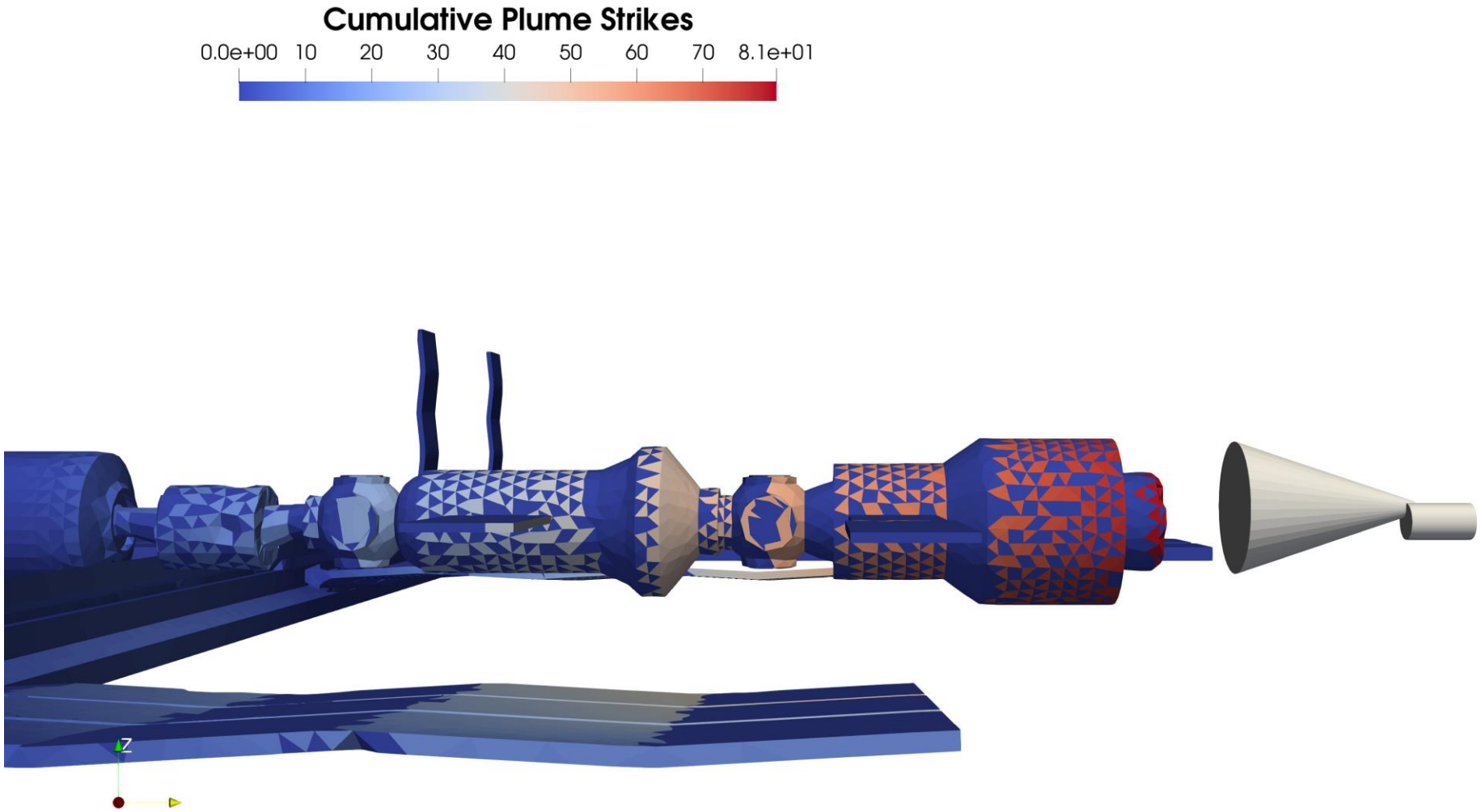




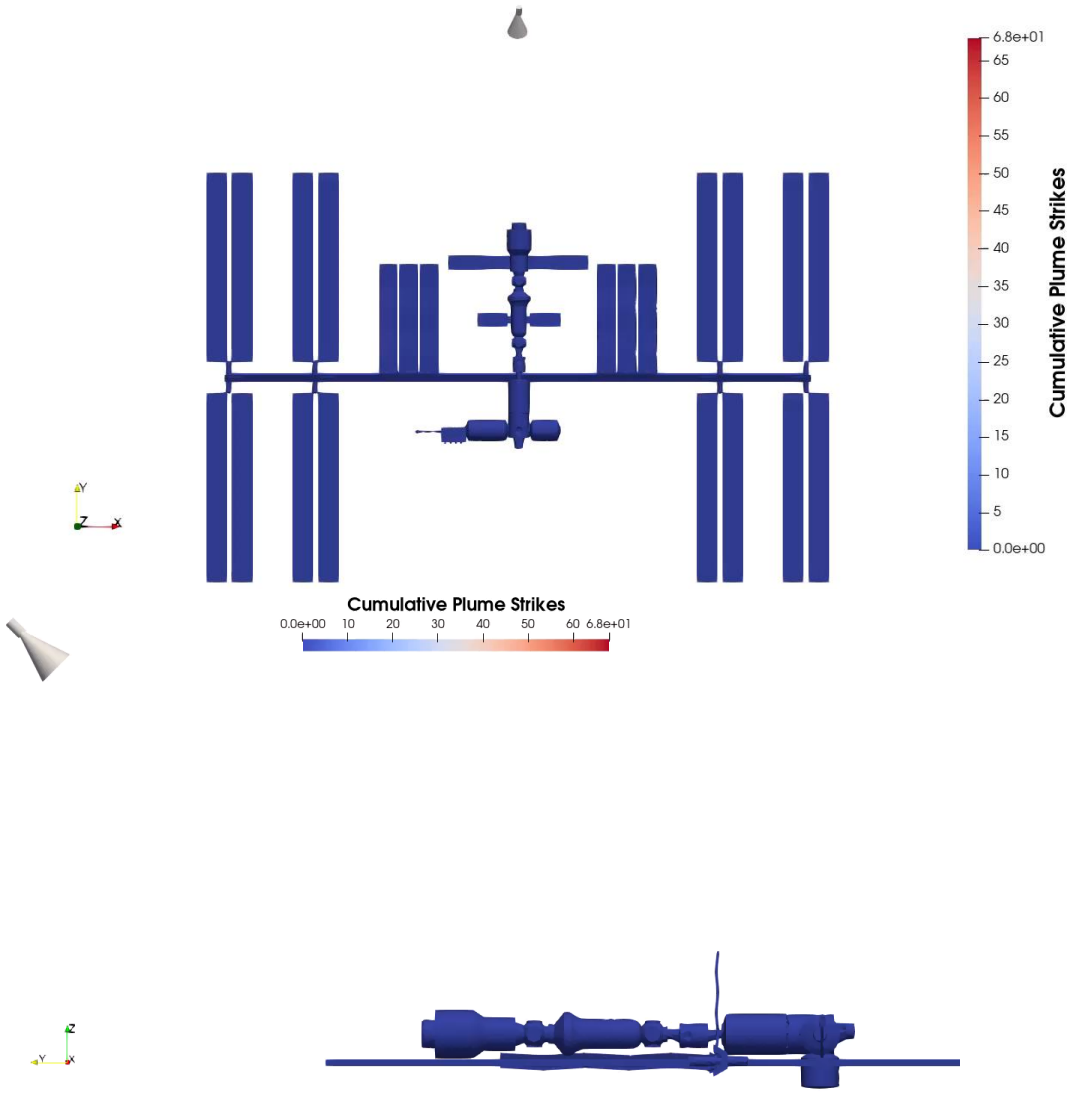
Demonstration Case: ISS Horizontal Docking



Demonstration Case: ISS Horizontal Docking

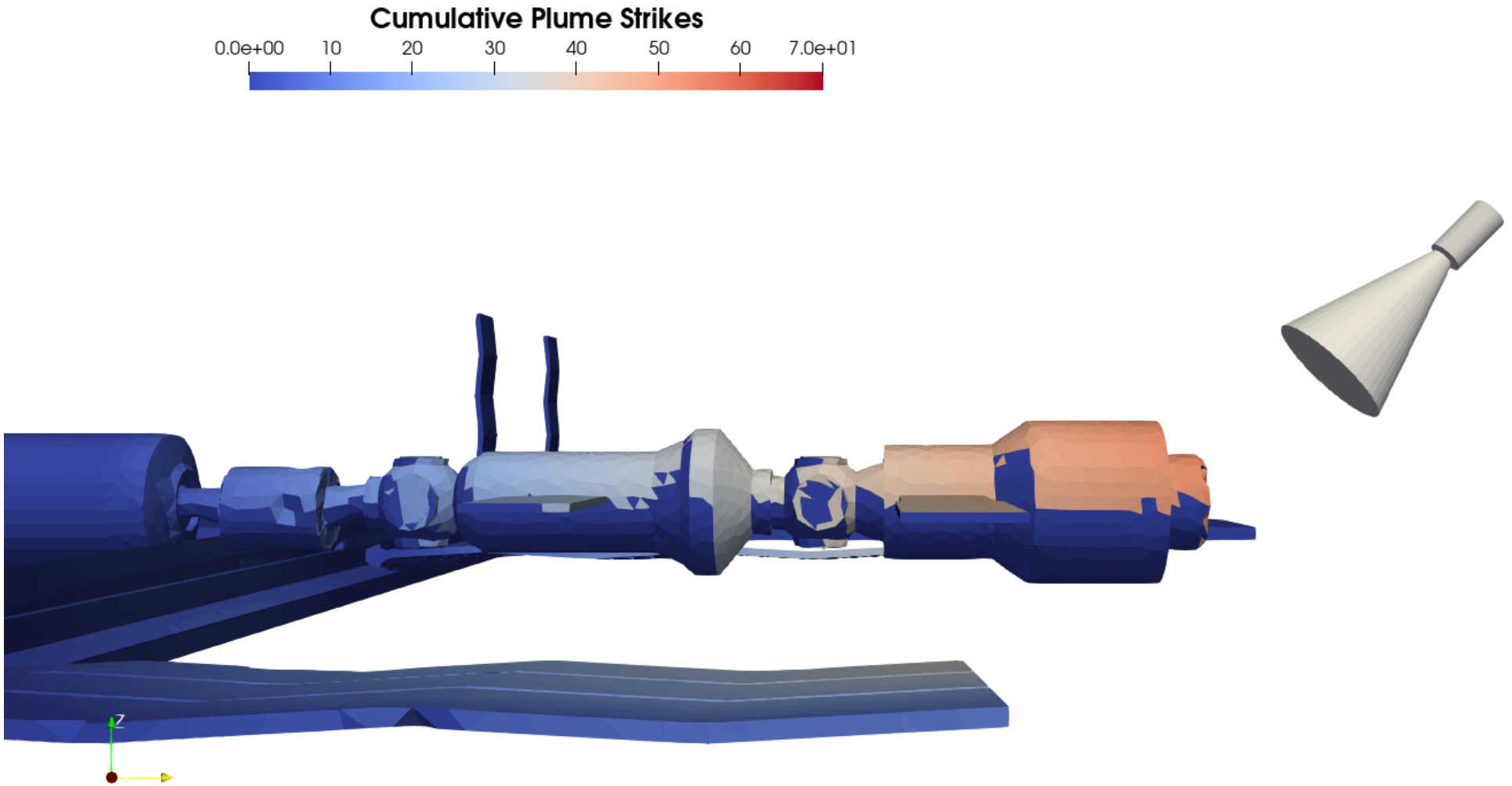


Demonstration Case: ISS Descending Line Docking

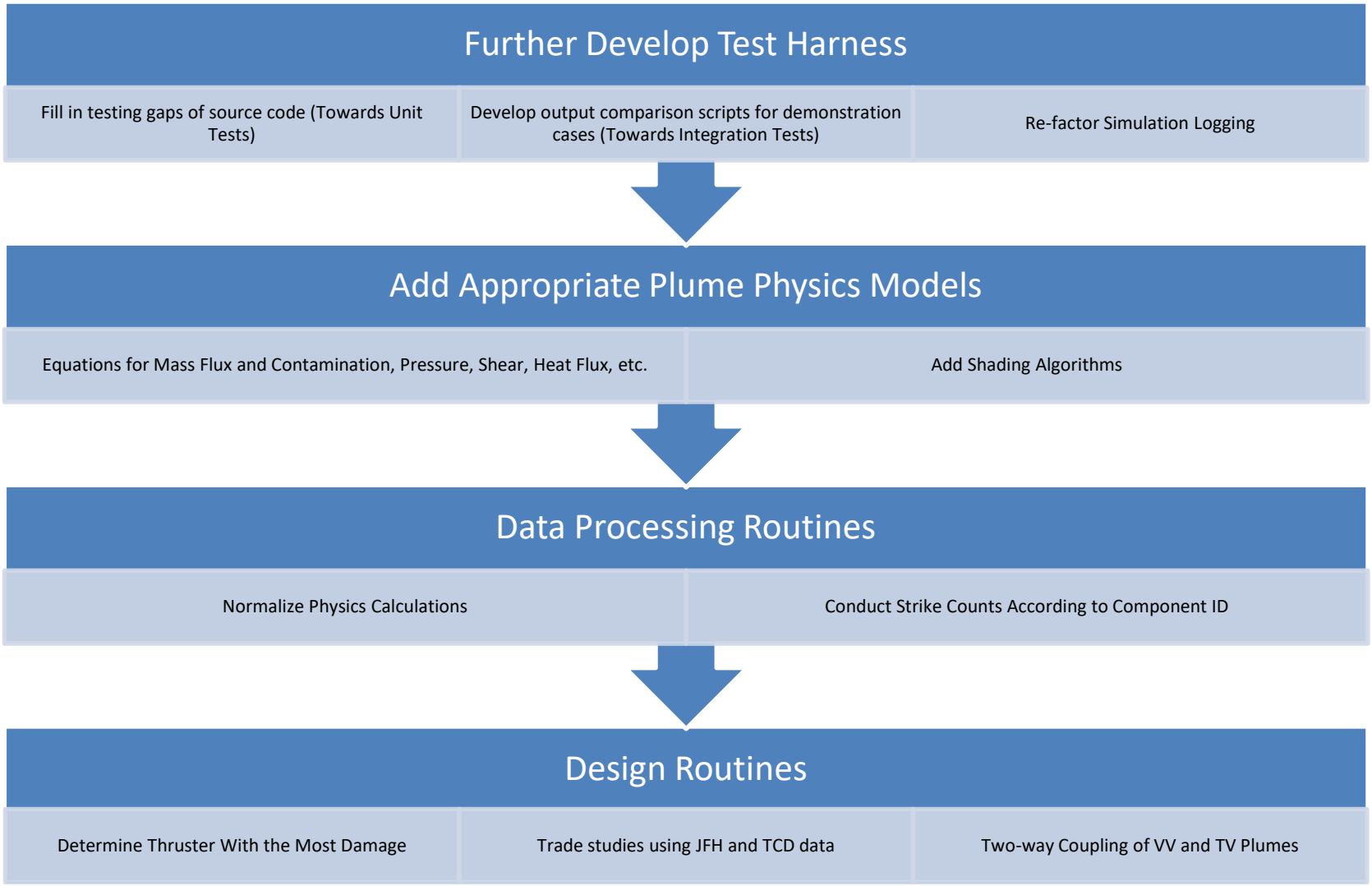


Plume Max Distance: 150 m
Plume Half-Angle: 45 degrees
Time Steps: 100
Run Time: 5 minutes

Demonstration Case: ISS Descending Line Docking



Next Steps


Further Develop Test Harness
Fill in testing gaps of source code (Towards Unit Tests)
Develop output comparison scripts for demonstration cases (Towards Integration Tests)
Re-factor Simulation Logging
Add Appropriate Plume Physics Models
Equations for Mass Flux and Contamination, Pressure, Shear, Heat Flux, etc.
Add Shading Algorithms
Data Processing Routines
Normalize Physics Calculations
Conduct Strike Counts According to Component ID
Design Routines
Determine Thruster With the Most Damage
Trade studies using JFH and TCD data
Two-way Coupling of VV and TV Plumes

Acknowledgements

Thank you to Florida Space Grant for funding this opportunity

Thank you to the O-Stem leaders for helping facilitate the internship

Thank you to Professor Michael Kinzel. First and continued supporter in CFD career.

Thank you to the DSL team for being so welcoming and willing to help us learn.

Thank you to Dr. Jonathan Pitt for the support and guidance over the past three semesters on studying plume impingement and much more!

Questions?

Thank You for Listening!
Any Questions?