

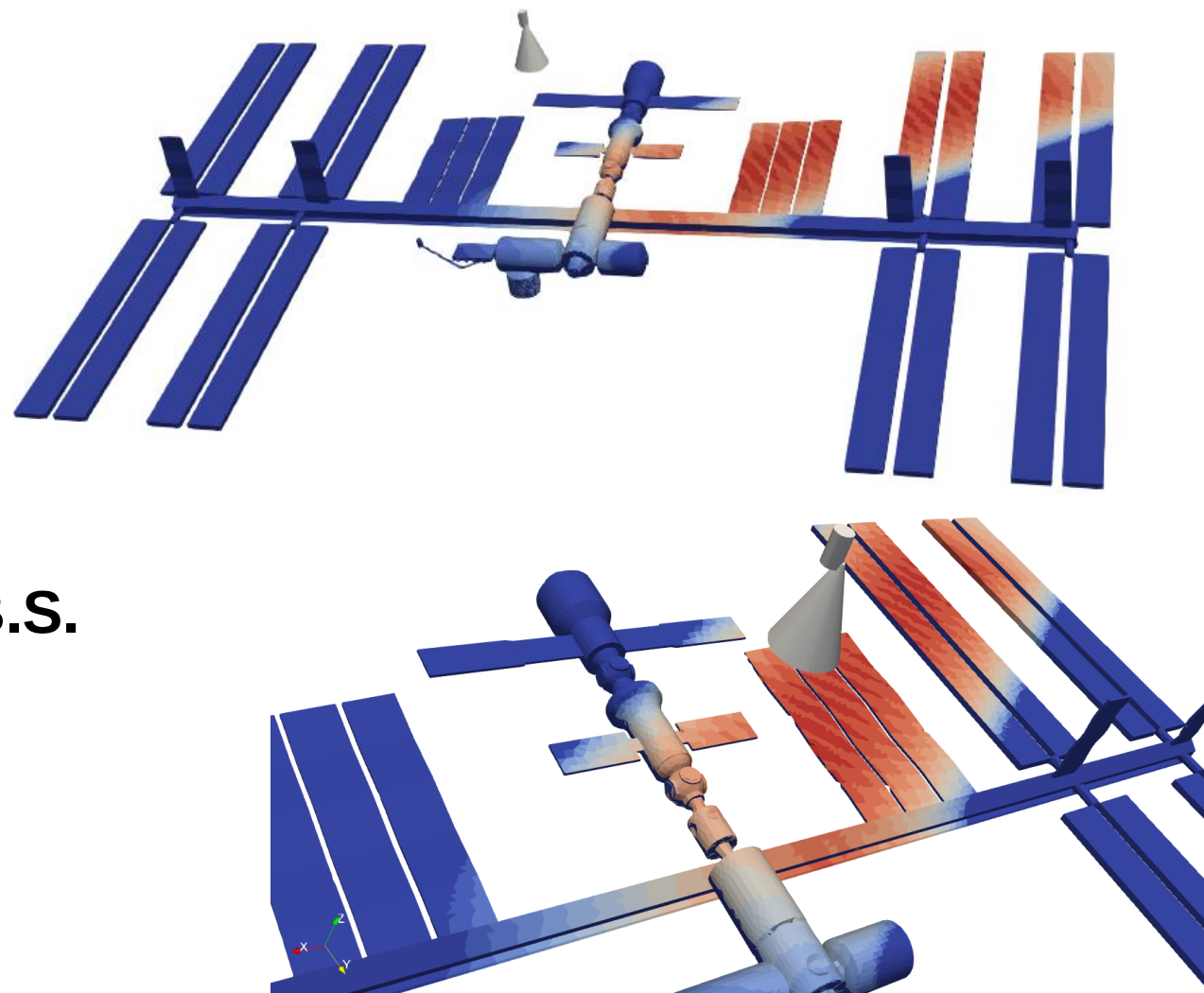


# Development Towards a Rapid Analysis Plume Dynamics Estimation Tool

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



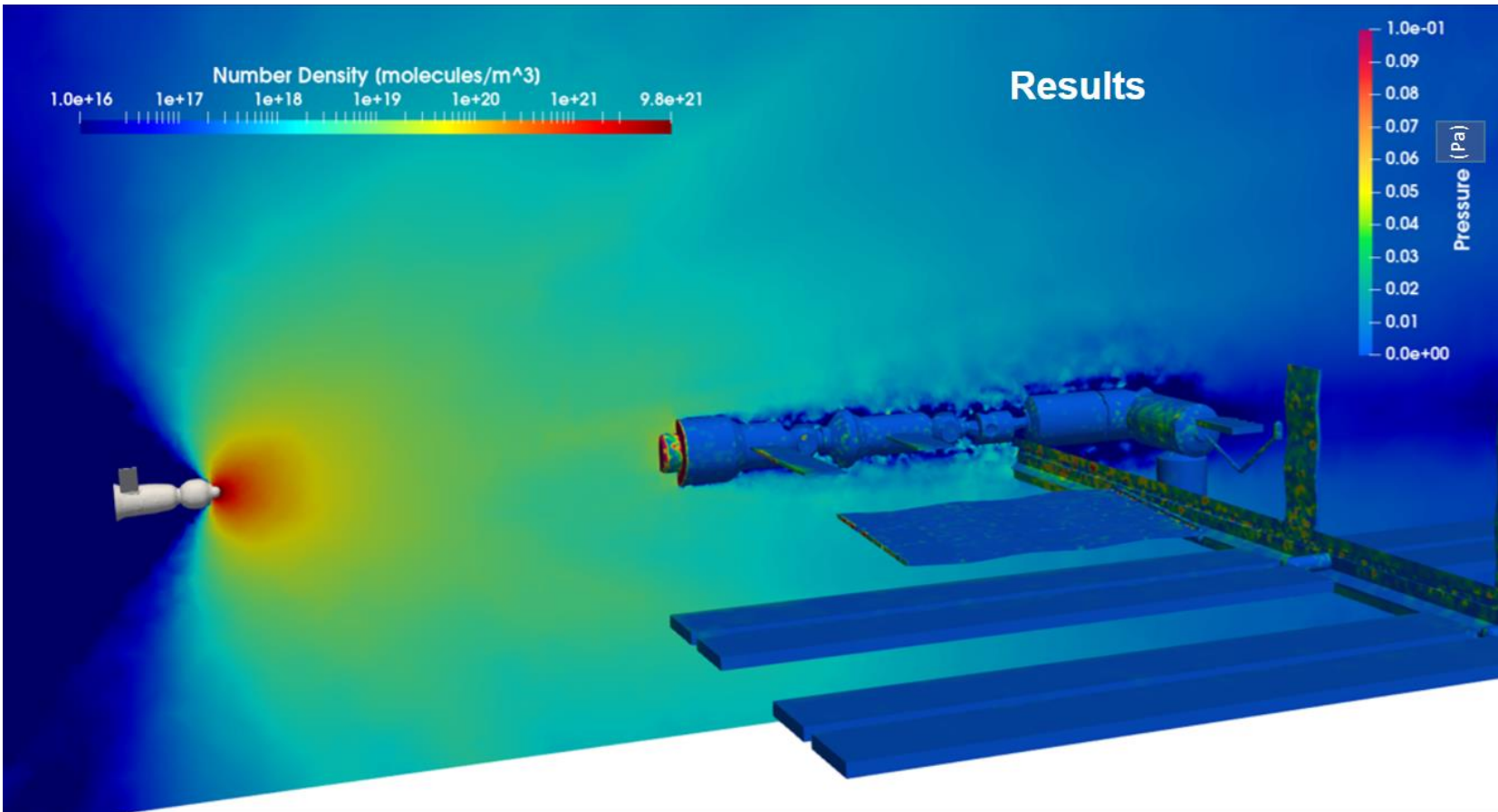


# Outline

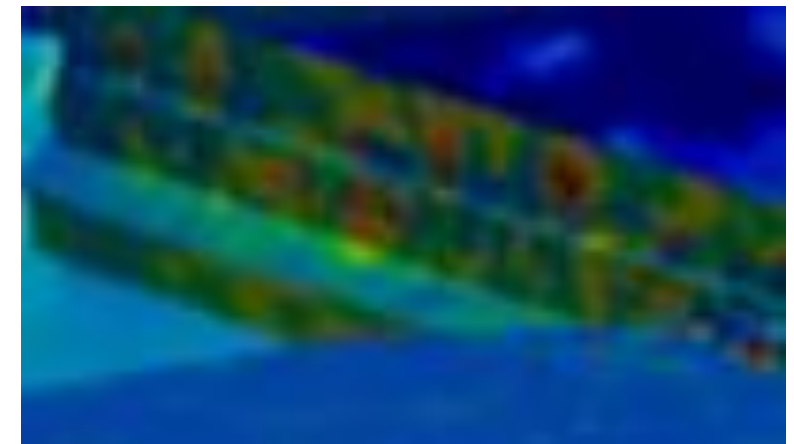


- **Background**
  - Previous Work and Motivation
- **Requirements**
  - Reading Inputs
  - Vehicle and Thruster Transformations
  - Plume Strike Logic
- **Demonstration Cases**
- **Future Steps**
- **Acknowledgements**

# Previous Work and Motivation



- Spring/Summer 2022 ran high fidelity Direct Simulation Monte Carlo (DSMC) calculations of rarefied plumes.
- Gives important and detailed impingement information but can be prohibitively expensive to run.
- **Thus, tools for executing engineering calculations are necessary for rapid estimation 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

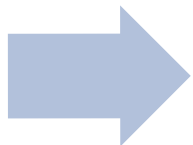
| Requirements                                                                                                                                                                    | Design Decision                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| 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 tranforming 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

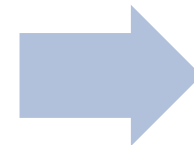
## Visiting Vehicle Data

- 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

File

wsl:localhost/Ubuntu-22.04/home/andy/Dev/dspset/doc/dspset/index.html

### Index

#### Sub-modules

- dspset.JetFiringHistory
- dspset.PlumeStrikeEstimationStudy
- dspset.PlumeStrikeModels
- dspset.TargetVehicle
- dspset.ThrusterConfigData
- dspset.Vehicle
- dspset.VisitingVehicle
- dspset.error
- dspset.file\_print
- dspset.output
- dspset.plume\_physics
- dspset.util

### Namespace **dspset**

#### Sub-modules

dspset.JetFiringHistory

Class used to read in data file for the Jet Firing History associated with an RPOD maneuver ...

dspset.PlumeStrikeEstimationStudy

This script executes a parametric study to asses plume strike damage from RPOD maneuvers ...

dspset.PlumeStrikeModels

Class used store different physics models used in the RPOD-PSE study ...

dspset.TargetVehicle

Class used to store STL and VTK data associated with the Target Vehicle. This class extends Vehicle which is used to load STL data ...

dspset.ThrusterConfigData

Class used to read in data file for the Thruster Configuration Data associated with a Visiting Vehicle ...

dspset.Vehicle

Class used to read in STL mesh data for Vehicles used in the an RPOD maneuver Plume Strike study.

dspset.VisitingVehicle

Class used to store STL and TCD data associated with the Visting Vehicle ...

dspset.error

dspset.file\_print

dspset.output

dspset.plume\_physics

dspset.util

Contains utility subroutines used across classes ...

## Version Control

```
andy@KSLW2023041007: ~/Dev/dspset/dspset
commit 8c6891b0643c81a9ff779434a4a06cd58b0f8d22a (HEAD -> master)
Author: Andres Torres-Figueroa <andres.j.torres-figueroa@nasa.gov>
Date: Sun Jul 23 12:53:40 2023 -0400

    updat test cases and data

commit 0184b87271a9a365d9b81dd35a1de79bc184ee39
Author: Andres Torres-Figueroa <andres.j.torres-figueroa@nasa.gov>
Date: Sun Jul 23 12:52:56 2023 -0400

    update docstrings, other comments, and remove un needed print statements

commit 8b460898b9f54657664d5689671874dbee81ef37
Author: Andres Torres-Figueroa <andres.j.torres-figueroa@nasa.gov>
Date: Sun Jul 23 10:24:15 2023 -0400

    update PSES to account for cumulative and plume strikes and strikes per firing

commit 3597dc89f64e88c212ad021493e73cc6d744d2e1
Author: Andres Torres-Figueroa <andres.j.torres-figueroa@nasa.gov>
Date: Sun Jul 23 10:07:35 2023 -0400

    fix typo in doc/README

commit d6676d4904204c31984b2eac2ea34d9f8bebebf6
Author: Andres Torres-Figueroa <andres.j.torres-figueroa@nasa.gov>
Date: Fri Jul 21 19:05:56 2023 -0400

    add data for iss_appr_quar_cir

commit 7d9b44296eb7748747c8af7f01de871666566561
Author: Andres Torres-Figueroa <andres.j.torres-figueroa@nasa.gov>
Date: Fri Jul 21 19:01:36 2023 -0400

    fix typo

commit 97a19387a1c4e144582c245d30e1ce87505d4c4f
Author: Andres Torres-Figueroa <andres.j.torres-figueroa@nasa.gov>
Date: Fri Jul 21 18:55:48 2023 -0400

    remove un-needed paarenter

commit 931cdfc35f849852fc2946f1e8455cbccc41b326
Author: Andres Torres-Figueroa <andres.j.torres-figueroa@nasa.gov>
Date: Fri Jul 21 18:52:34 2023 -0400

    add test case files for the iss_quar_cir_appr

commit c4c1ad9f9ce60d55ff3d20f8031afc967a5b483
Author: Andres Torres-Figueroa <andres.j.torres-figueroa@nasa.gov>
Date: Fri Jul 21 18:41:55 2023 -0400

    refactor for visibility

commit 999e6aaabd3c0b8a3018d7db403cf12f68fe9d8
Author: Andres Torres-Figueroa <andres.j.torres-figueroa@nasa.gov>
Date: Mon Jul 17 09:07:35 2023 -0400

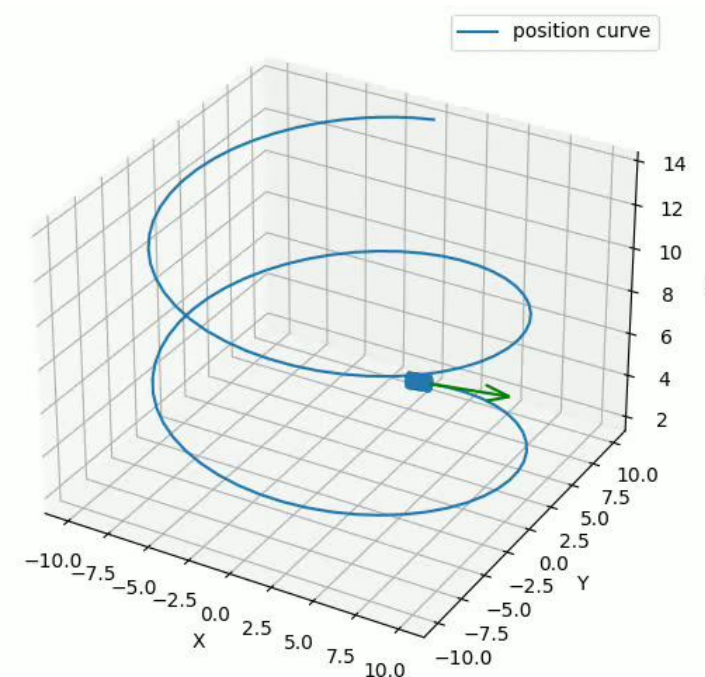
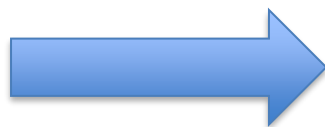
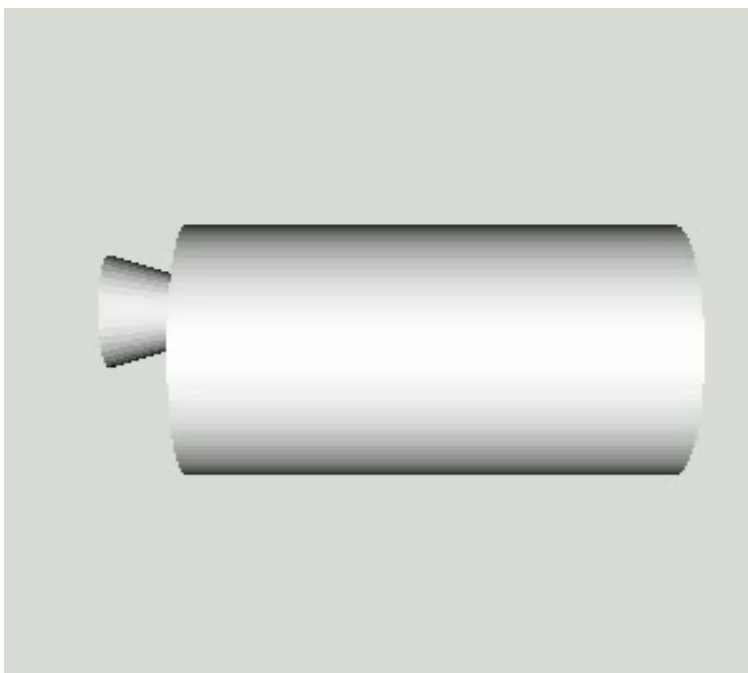
    update PSET class to save strikes and plume rotations: used flat_plate_fly_by to demonstrate

commit 768ade8a06d7f3876fab8034e9ba52159389f456
Author: Andres Torres-Figueroa <andres.j.torres-figueroa@nasa.gov>
Date: Mon Jul 17 09:00:31 2023 -0400

    stl/ --> add new stl resources

commit 9ebc1a73675619645d124819812c10ec62502d
Author: Andres Torres-Figueroa <andres.j.torres-figueroa@nasa.gov>
:
```

# Vehicle and Thruster Transformations



$$\begin{bmatrix} \hat{x}_{VV} \\ \hat{y}_{VV} \\ \hat{z}_{VV} \end{bmatrix} = \begin{bmatrix} VV_{11} & VV_{21} & VV_{31} \\ VV_{12} & VV_{22} & VV_{32} \\ VV_{13} & VV_{23} & VV_{33} \end{bmatrix}_{VV} \begin{bmatrix} \hat{x}_i \\ \hat{y}_i \\ \hat{z}_i \end{bmatrix}$$

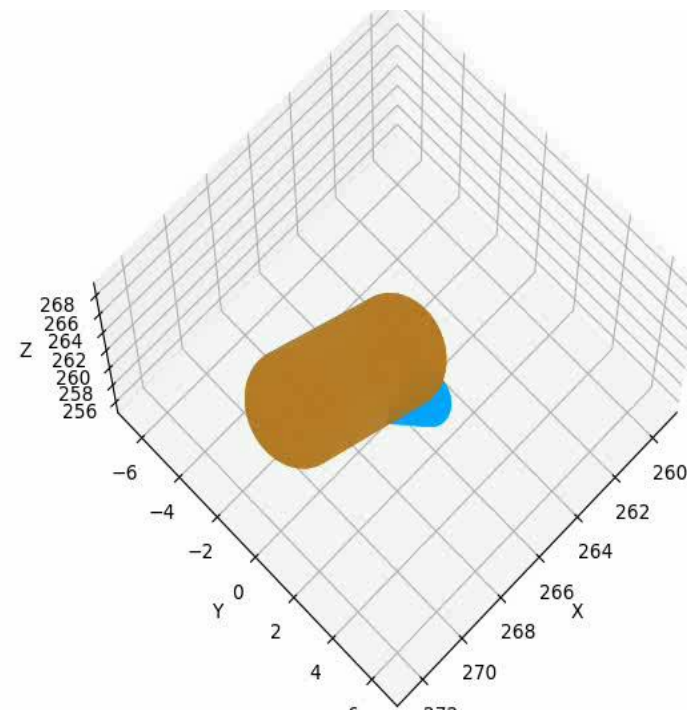
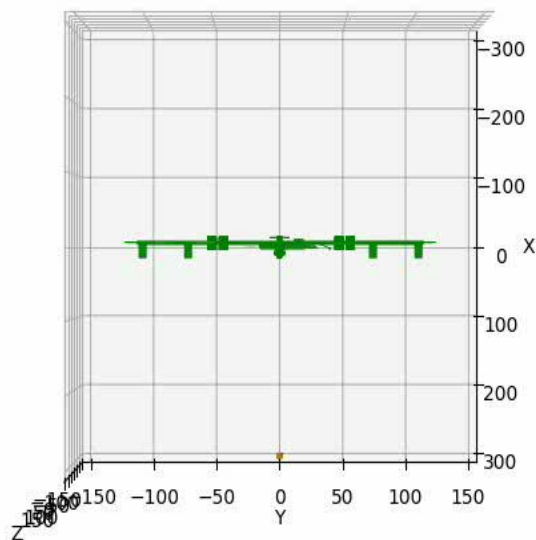
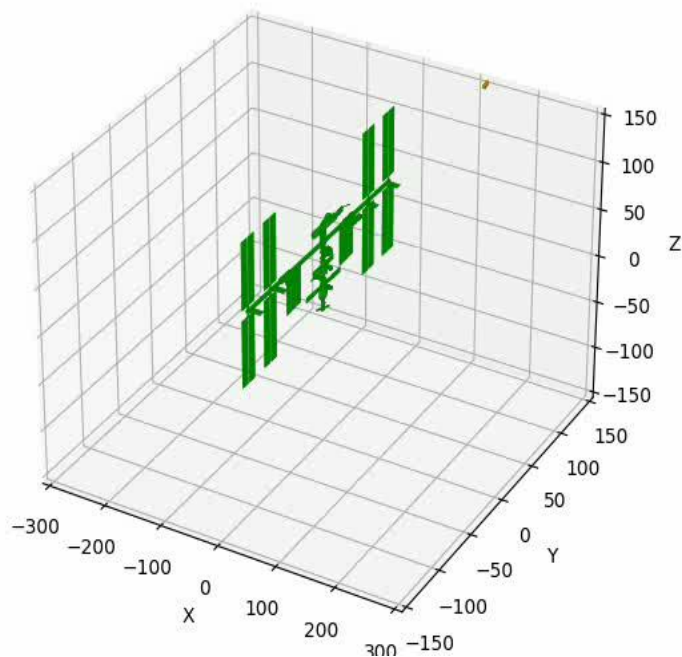
Specified For Each Thruster Using  
Thruster Configuration Data

$$\begin{bmatrix} \hat{x}_{TV} \\ \hat{y}_{TV} \\ \hat{z}_{TV} \end{bmatrix} = \begin{bmatrix} TV_{11} & TV_{21} & TV_{31} \\ TV_{12} & TV_{22} & TV_{32} \\ TV_{13} & TV_{23} & TV_{33} \end{bmatrix}_{TV} \begin{bmatrix} \hat{x}_{VV} \\ \hat{y}_{VV} \\ \hat{z}_{VV} \end{bmatrix}$$

Specified For Each Firing Using  
Jet Firing History

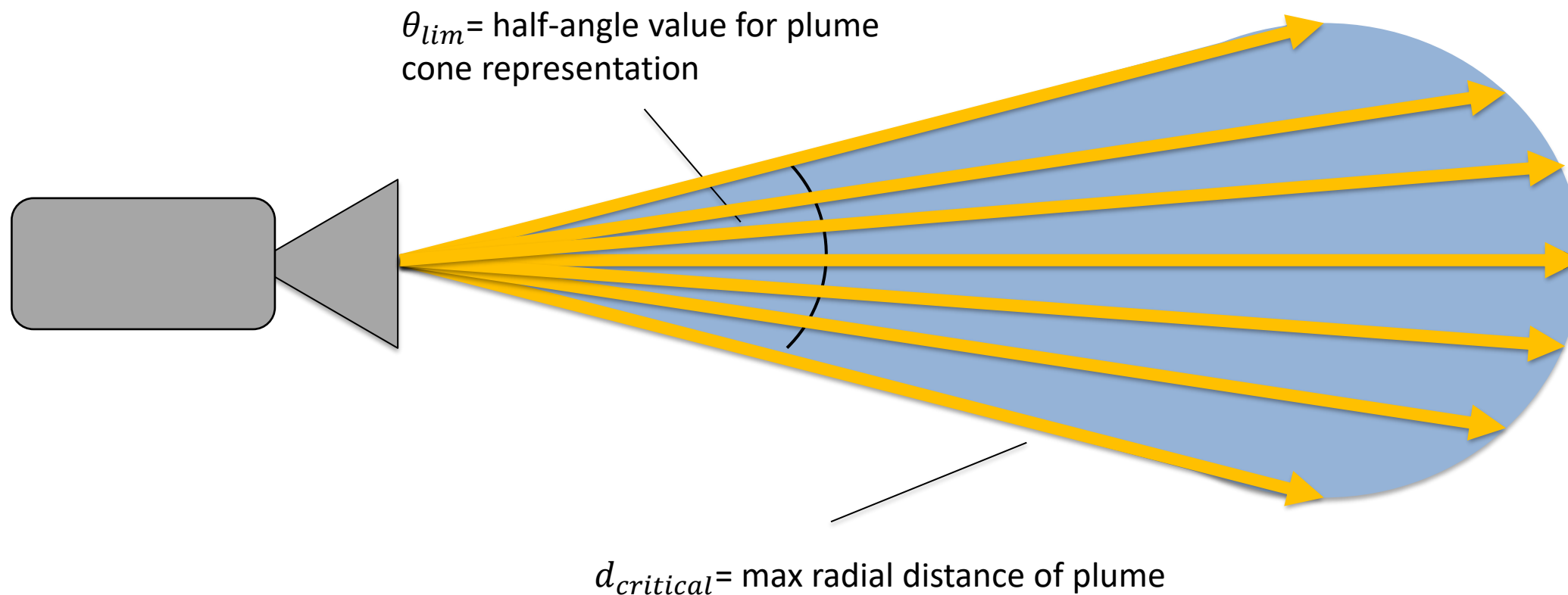


# Vehicle and Thruster Transformations



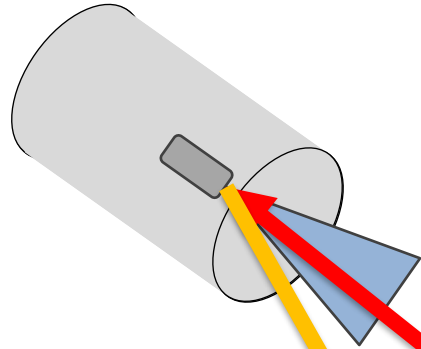
# Plume Cone Configuration

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



# Plume Strike Logic

Distance Vector from RCS Thruster to Surface Elements of Target Vehicle

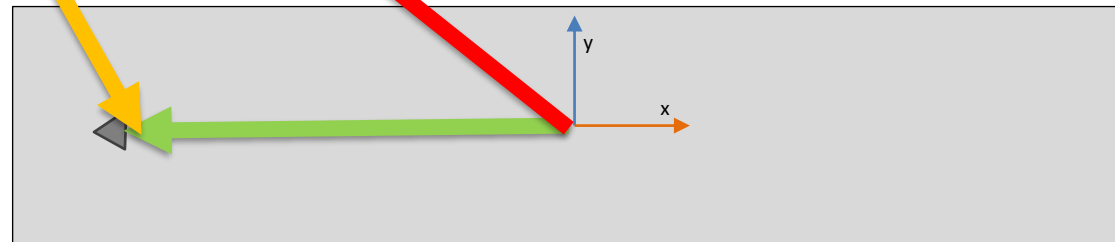


→  $\vec{r}_t$  = position vector for thruster exit wrt to TV

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

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

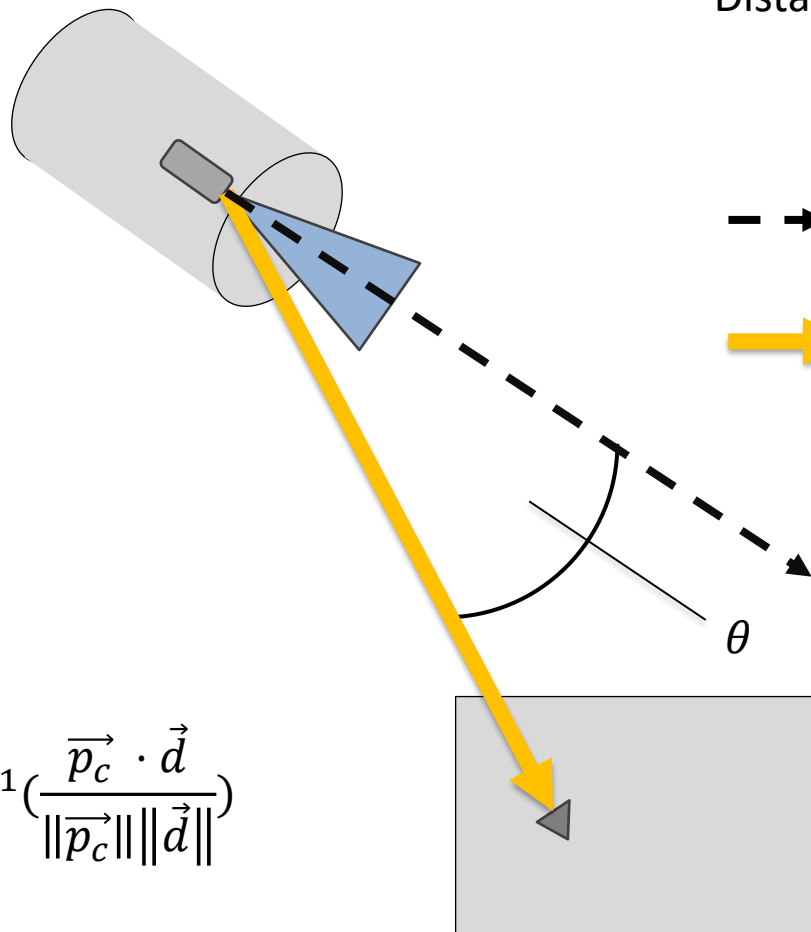
$$\vec{d} = \vec{r}_f - \vec{r}_t$$



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

# Plume Strike Logic

Angle Between Plume Centerline and  
Distance Vector to STL element



$\vec{p}_c$  = plume center line vector

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

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

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

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

# Plume Strike Logic

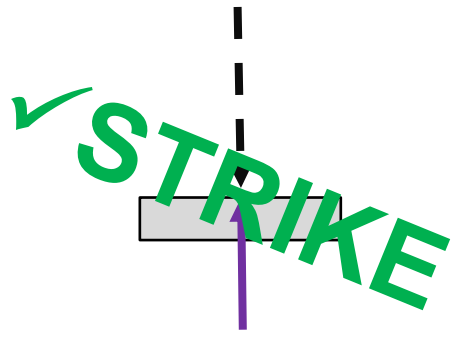
Face Orientation Check:  
Pre-Cursor to Shading  
Calculations

Dot Product  
Evaluation

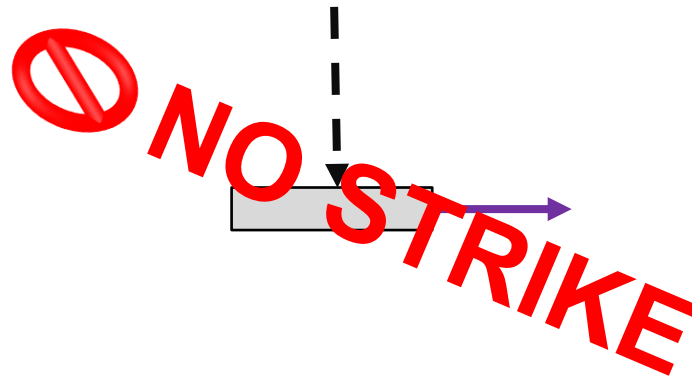
$$\vec{p}_c \cdot \vec{n}$$

— →  $\vec{p}_c = \text{plume center line}$

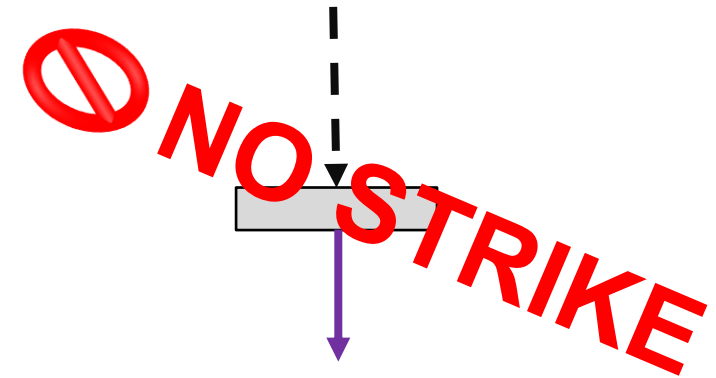
→  $\vec{n} = \text{normal vector for STL element}$



*Facing Each Other*  
 $\vec{p}_c \cdot \vec{n} = -1$



*Orthogonal*  
 $\vec{p}_c \cdot \vec{n} = 0$



*Facing Same Direction*  
 $\vec{p}_c \cdot \vec{n} = 1$



# 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

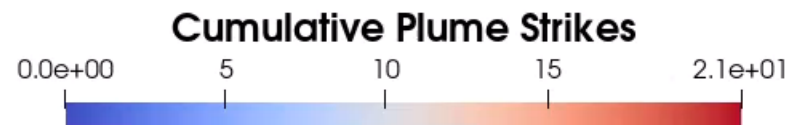
**else**  
    do not add a strike to the current stl element



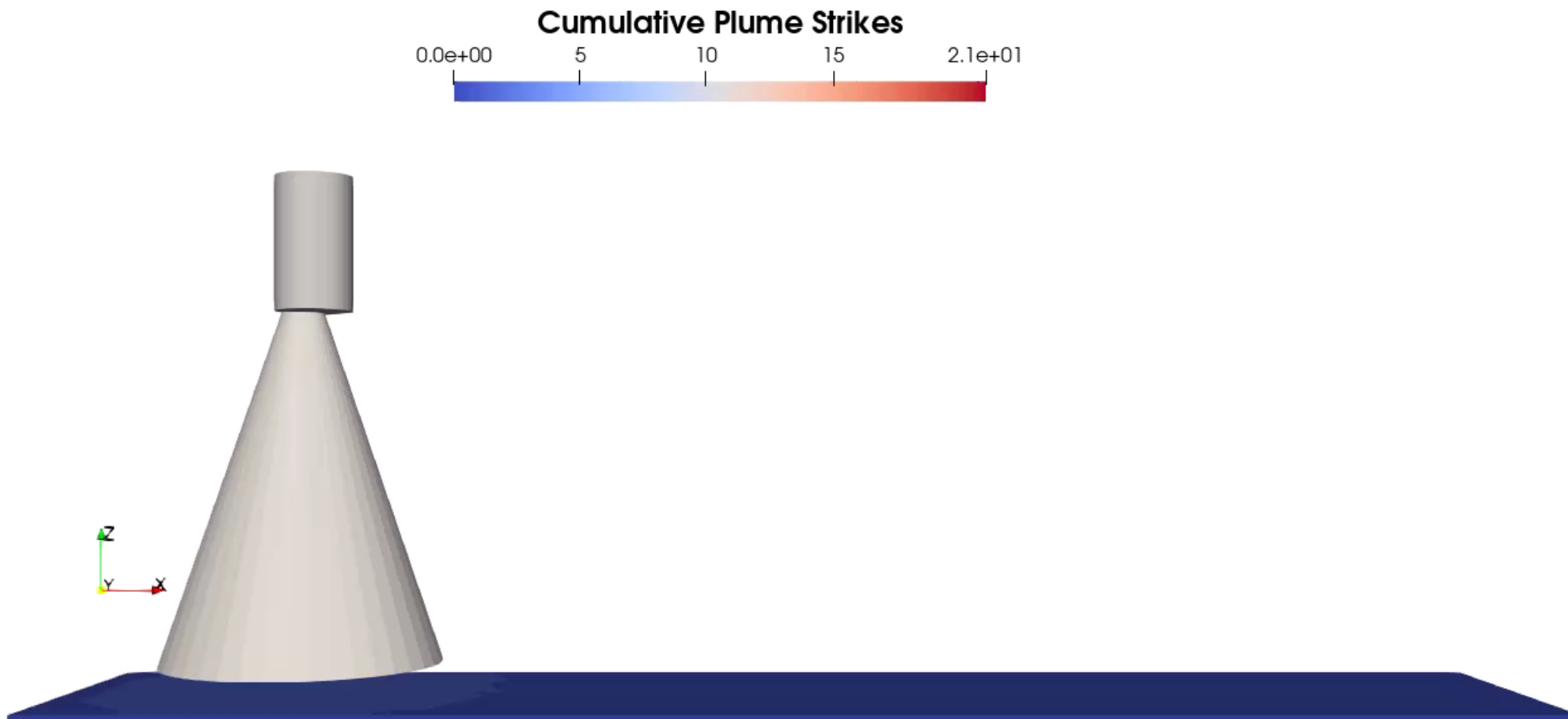
# Demonstration Case: Flat Plate Fly By



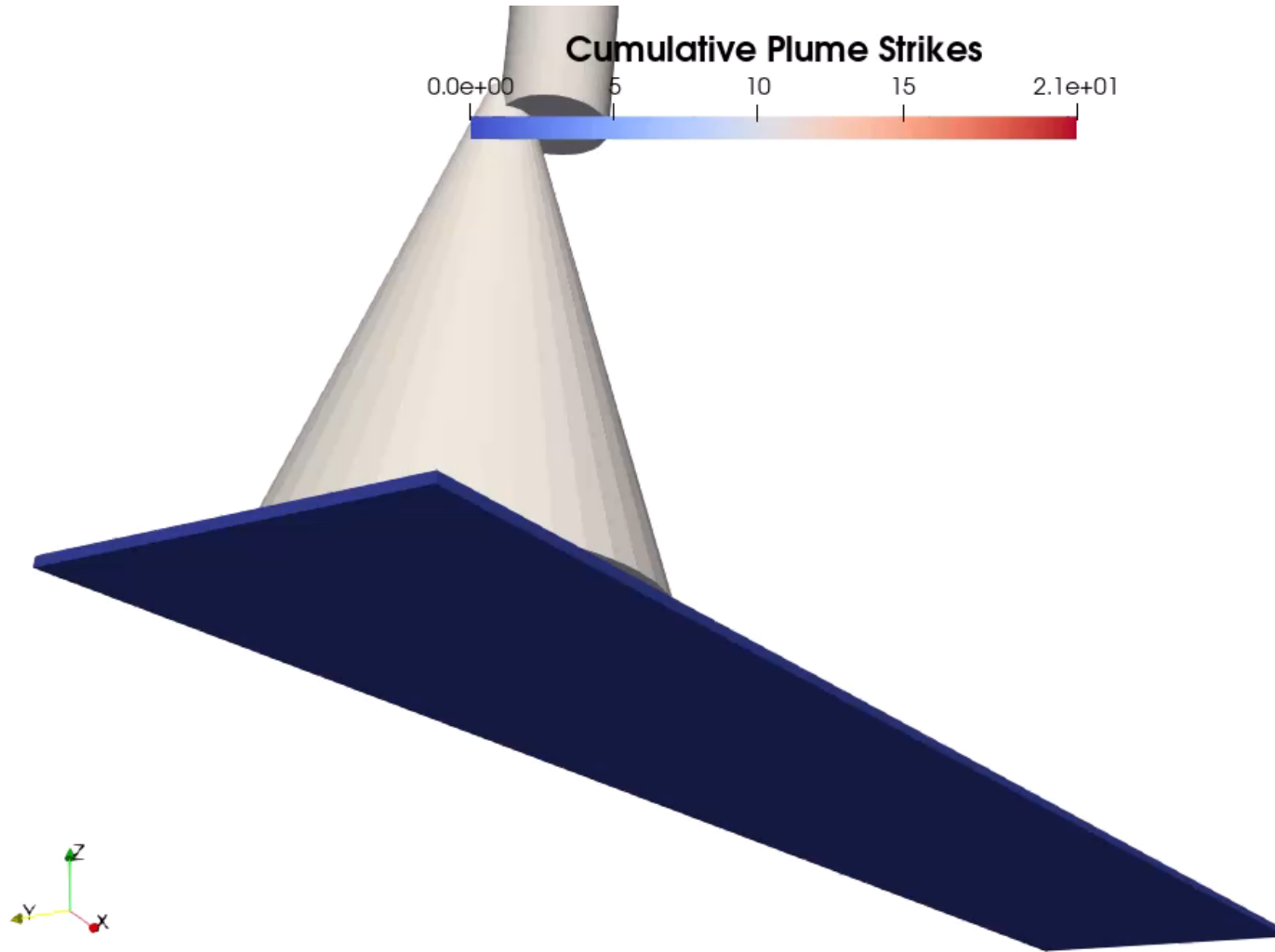
Plume Max Distance: 25 m  
Plume Half-Angle: 25 degrees  
Time Steps: 50  
Run Time: 9 seconds



# Demonstration Case: Flat Plate Fly By

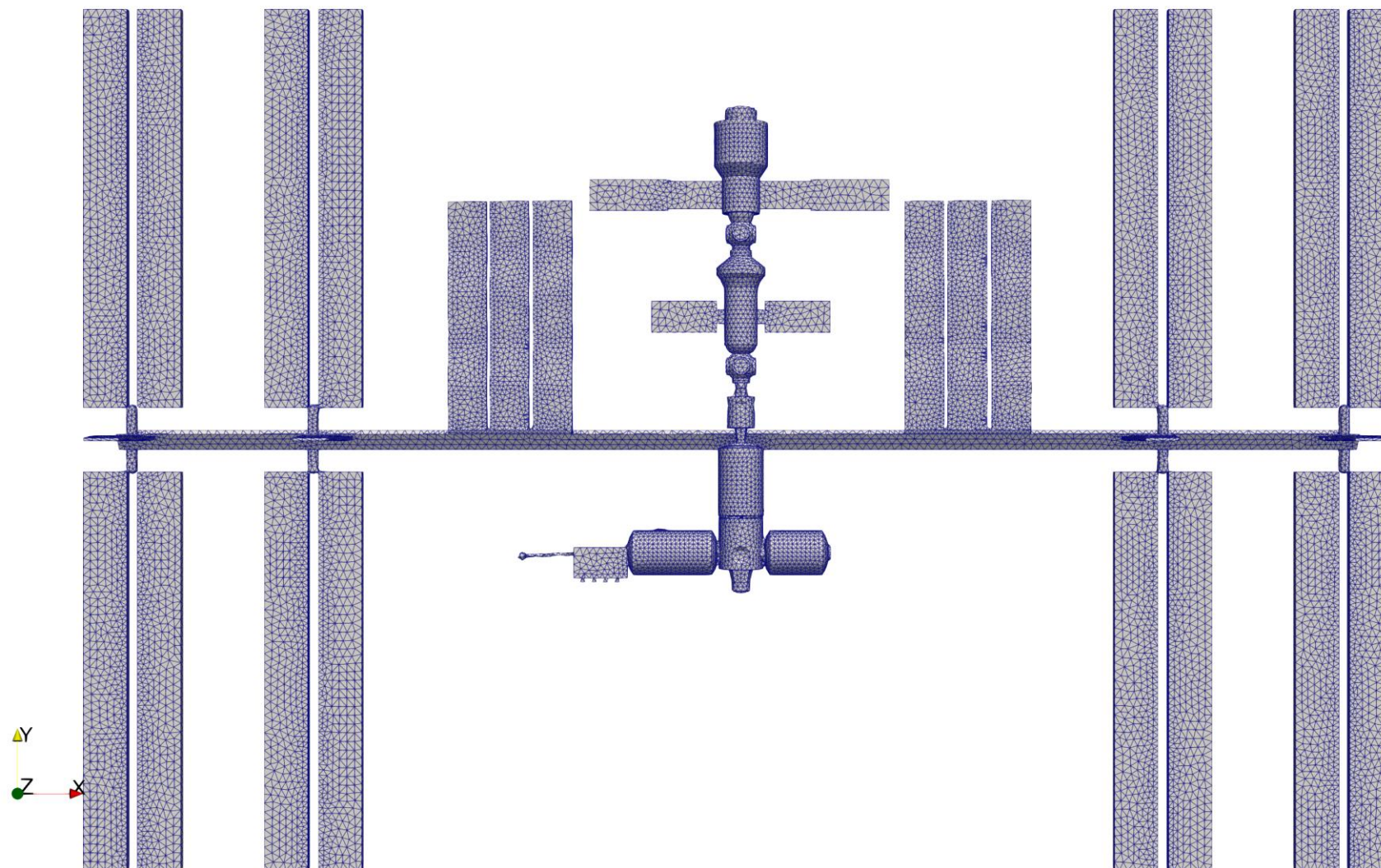


# Demonstration Case: Flat Plate Fly By





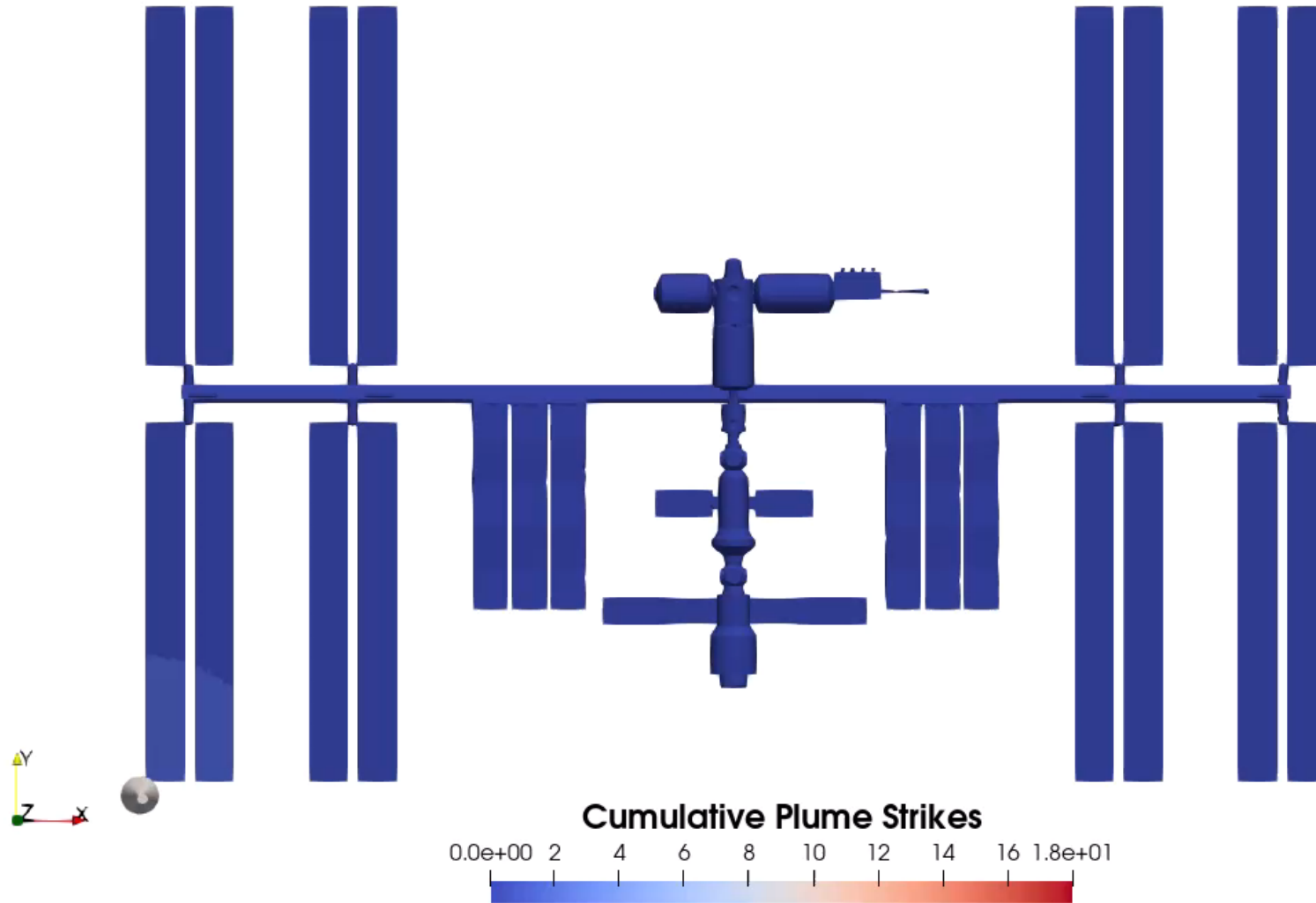
# ISS STL Surface



Cell Count: 75,000 faces



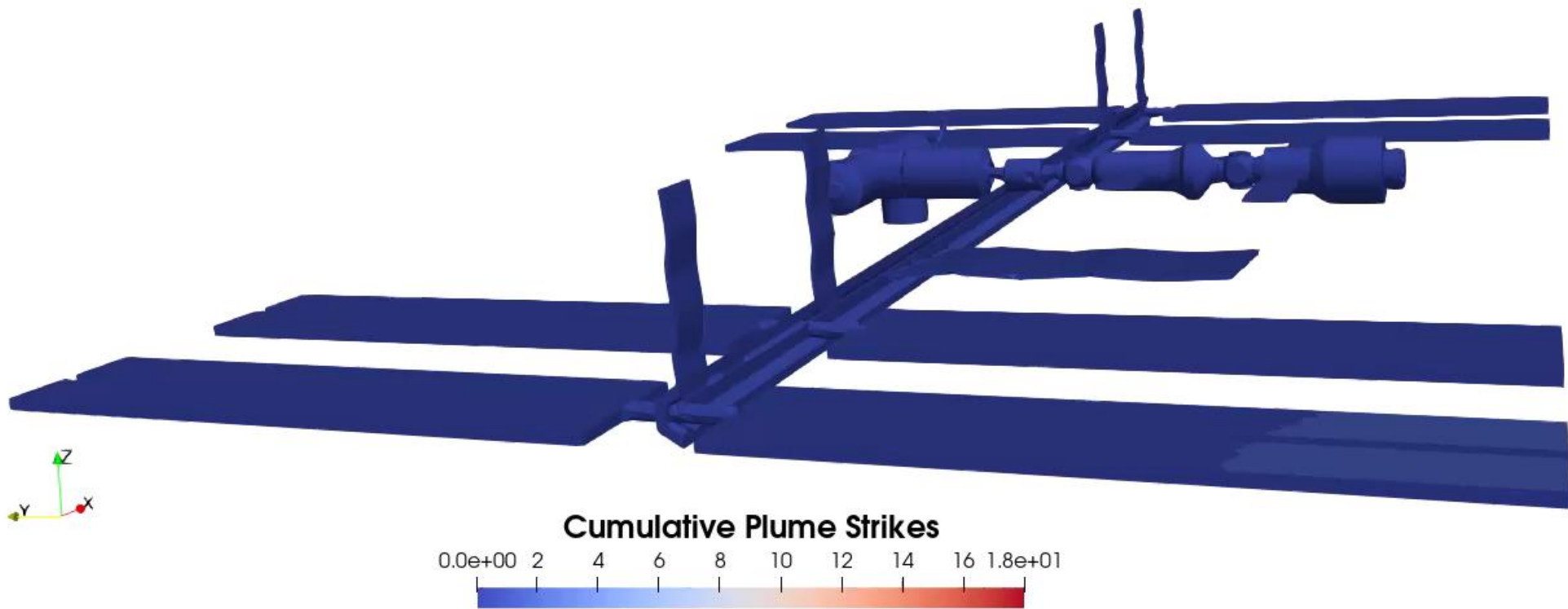
# Demonstration Case: ISS Fly By



Time Steps: 100  
Run Time: 5 minutes



# Demonstration Case: ISS Fly By

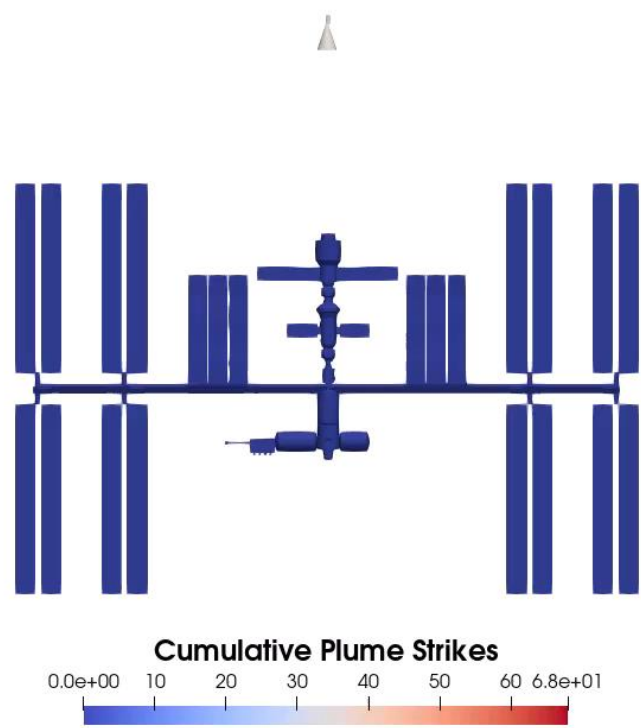
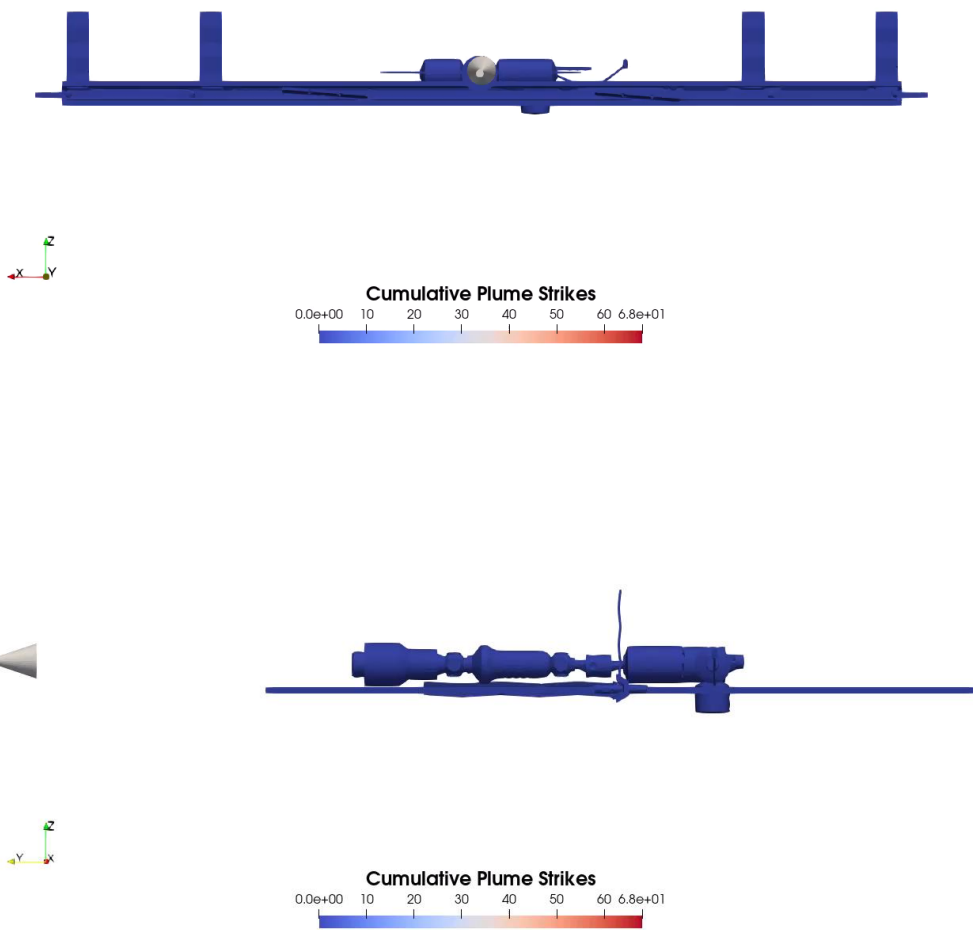




# Demonstration Case: ISS Horizontal Docking

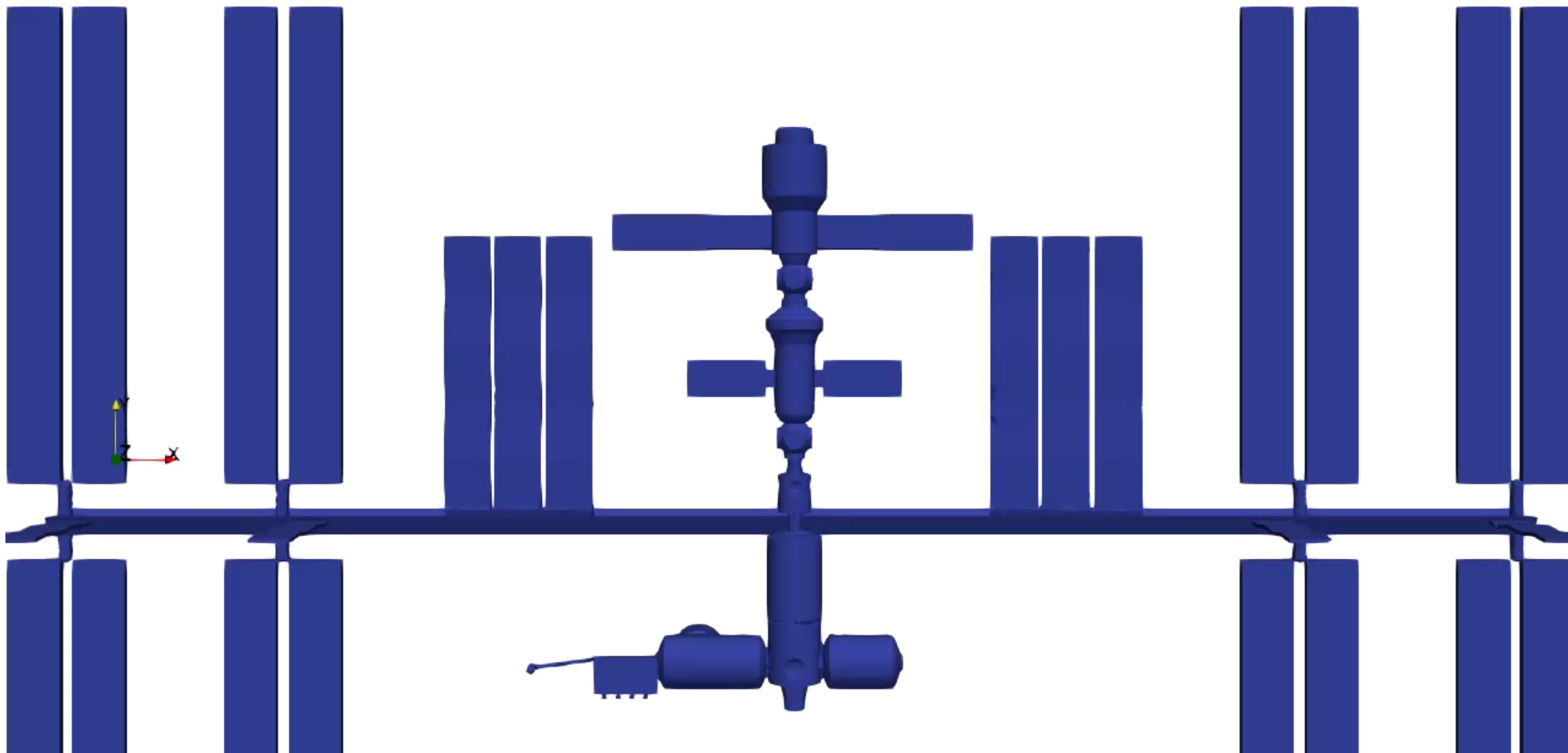


Plume Max Distance: 150 m  
Plume Half-Angle: 45 degrees  
Time Steps: 100  
Run Time: 5 minutes





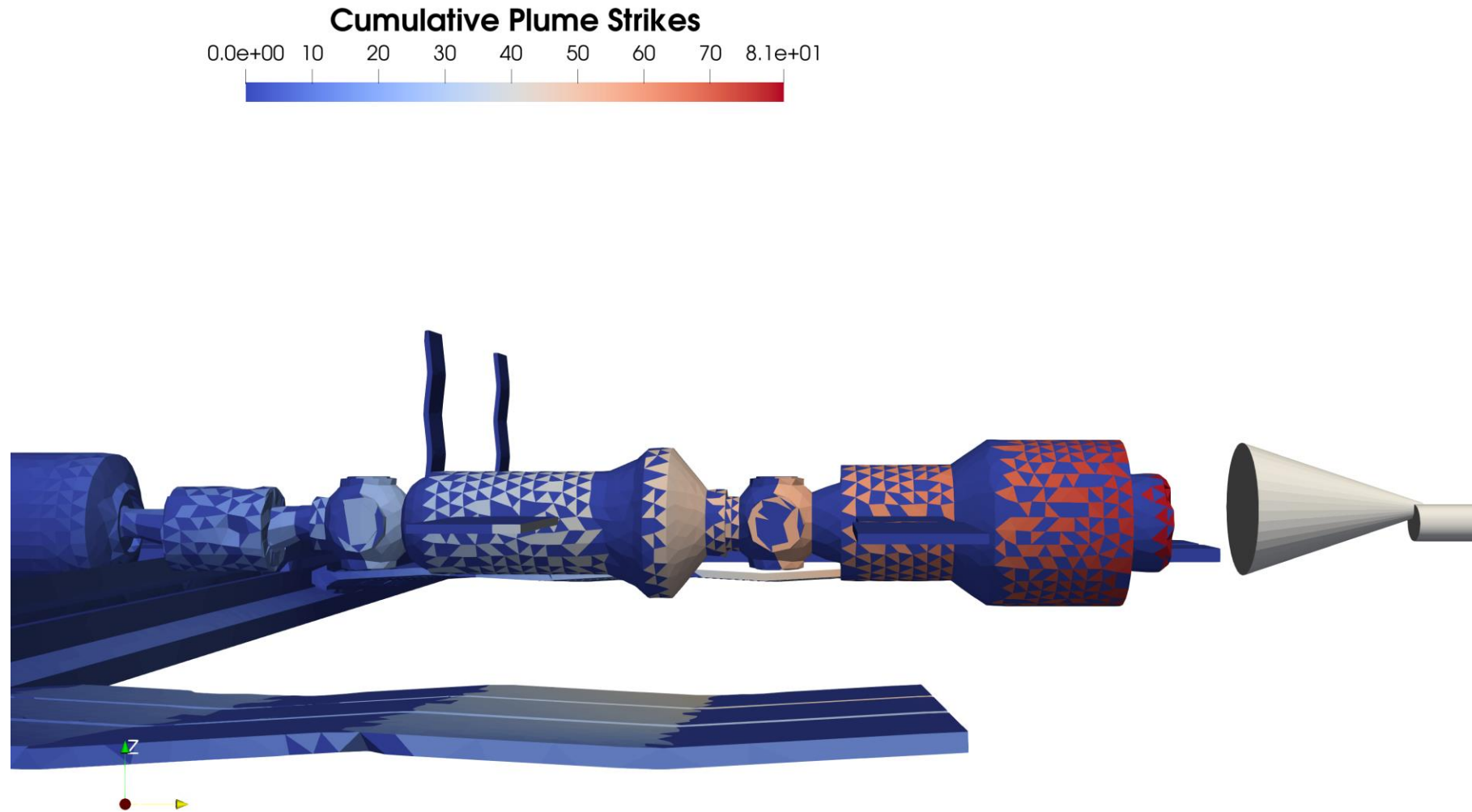
# Demonstration Case: ISS Horizontal Docking



Deep Space Logistics

FLY | SUPPLY | EXPLORE

# Demonstration Case: ISS Horizontal Docking

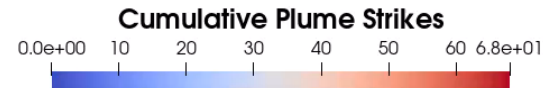
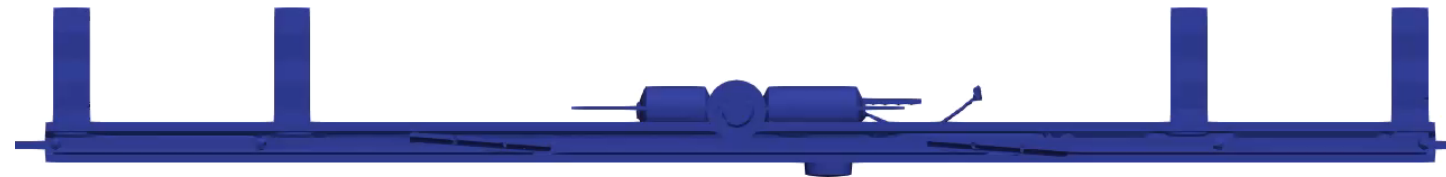
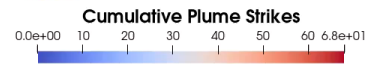
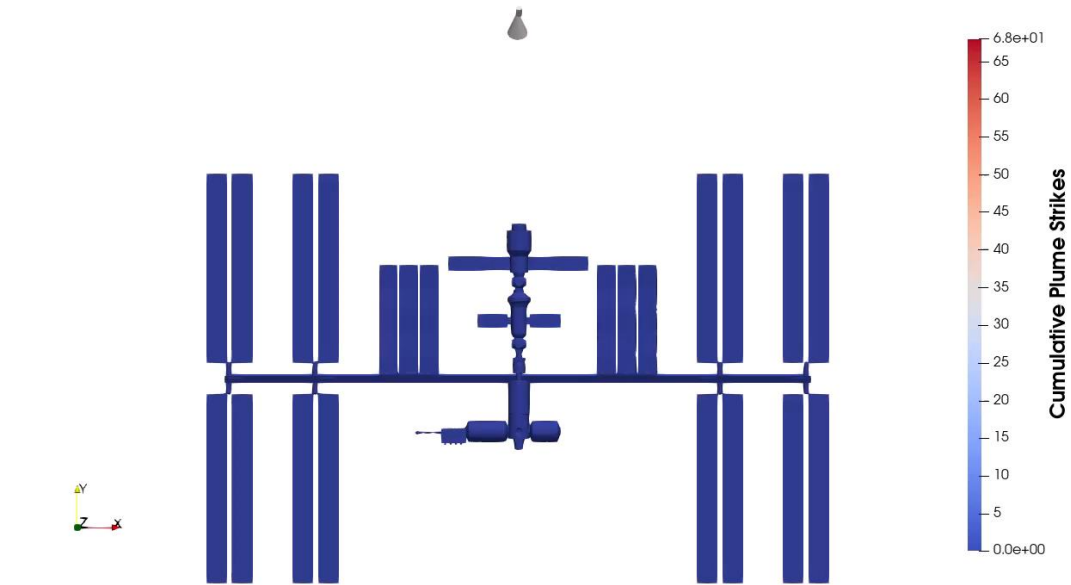




# Demonstration Case: ISS Descending Line Docking



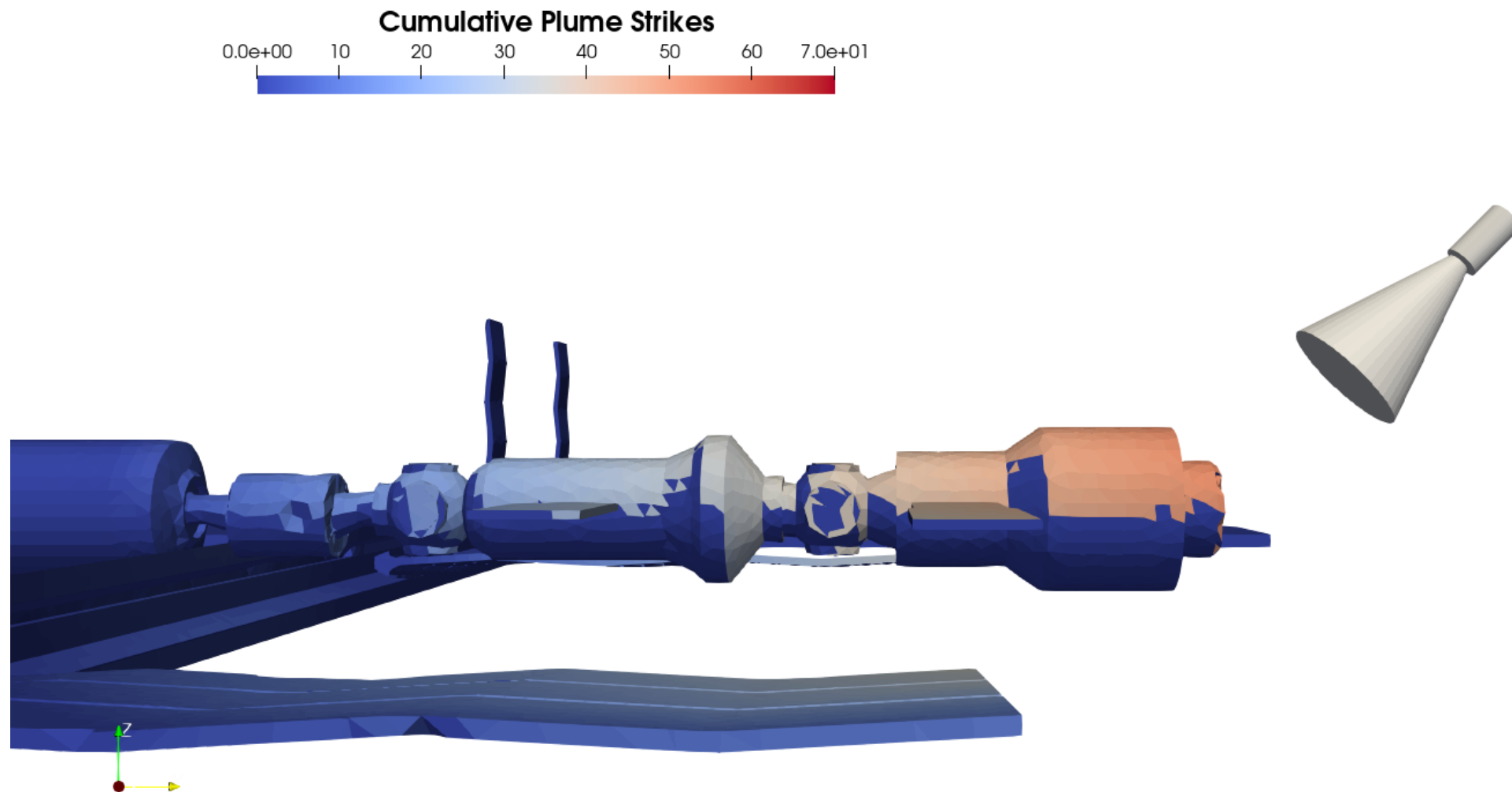
Plume Max Distance: 150 m  
Plume Half-Angle: 45 degrees  
Time Steps: 100  
Run Time: 5 minutes



Deep Space Logistics

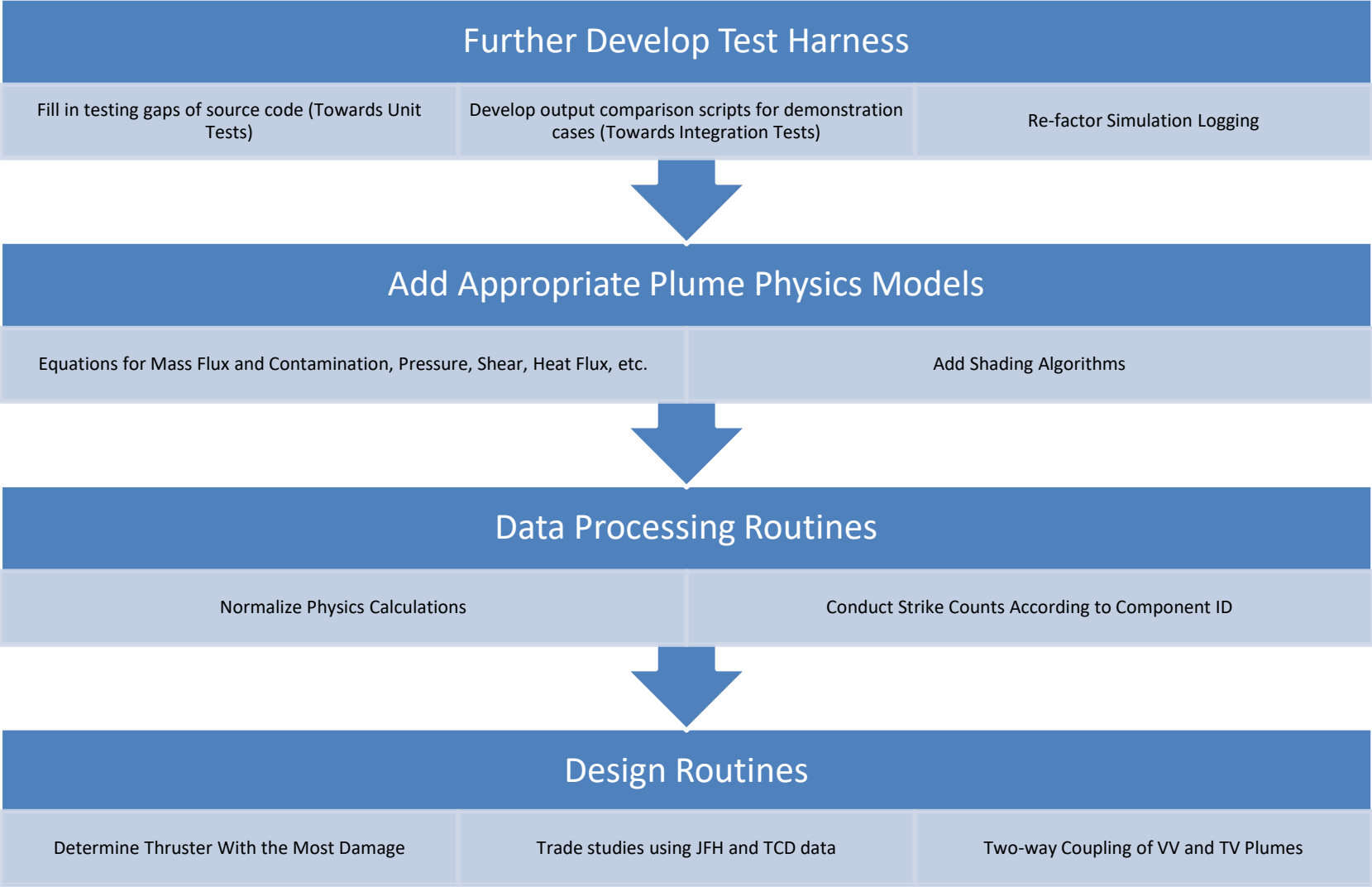
FLY | SUPPLY | EXPLORE

# Demonstration Case: ISS Descending Line Docking





# Next Steps





# 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?**