NREL
NATIONAL RENEWABLE ENERGY LABORATORY

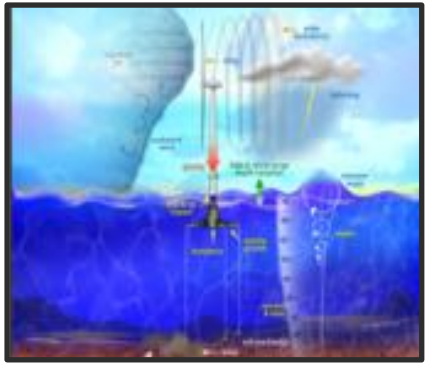
Starting with WISDEM

The Short Course

NREL

NREL has both turbine- and plant/system-focused research programs

Turbine Focused

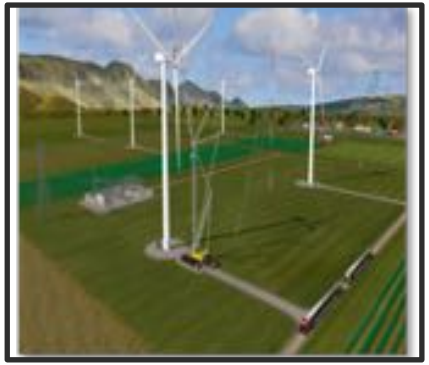


OpenFAST

Computer engineering tool for simulating the coupled dynamic response of wind turbines

Flagship software product

System Focused

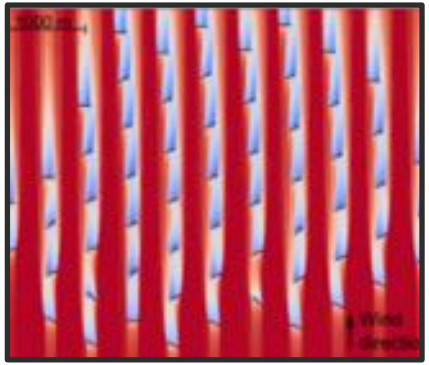


WISDEM™

Wind-plant Integrated System Design and Engineering Model

FOCUS

Farm Focused



FLORIS

A controls-oriented engineering wake model

FAST.Farm

WindSE

NALU

FOCUS

Agenda

- (Brief) Introduction to WISDEM and OpenMDAO
- WISDEM installation (essentials only)
- Tutorial 1: Run a simple WISDEM calculation with Jupyter Notebooks
- Some more detail on OpenMDAO
- Tutorial 2: Finding the Betz Limit through OpenMDAO optimization
- ~~Tutorial 3: Sellar problem for putting multiple models together~~
- Tutorial 4: Modeling a whole turbine and plant LCOE

(Brief) Introduction to WISDEM and OpenMDAO

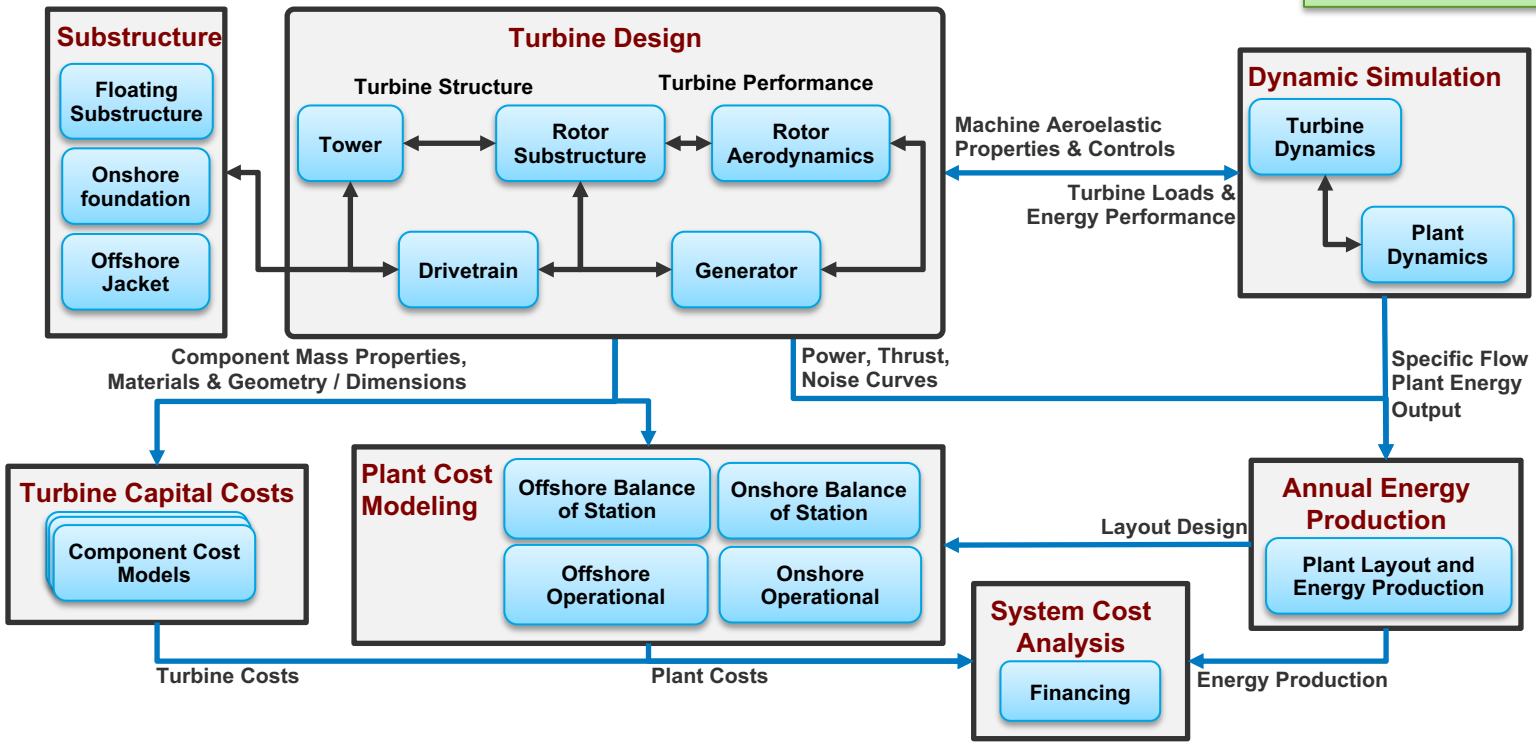
- **(Brief) Introduction to WISDEM and OpenMDAO**
- WISDEM installation (essentials only)
- Tutorial 1: Run a simple WISDEM calculation with Jupyter Notebooks
- Some more detail on OpenMDAO
- Tutorial 2: Finding the Betz Limit through OpenMDAO optimization
- Tutorial 4: Modeling a whole turbine and plant

WISDEM: Creates a virtual, vertically integrated wind plant from components to operations

WISDEM: Wind-Plant Integrated System Design & Engineering Model

- Integrated turbine design (e.g. rotor aero-structure, full turbine optimization)
- Integrated plant design and operations (e.g. wind plant controls and layout optimization)
- Integrated turbine and plant optimization (e.g. site-specific turbine design)

Modular design allows "plug-and-play" with external (3rd party) component modules

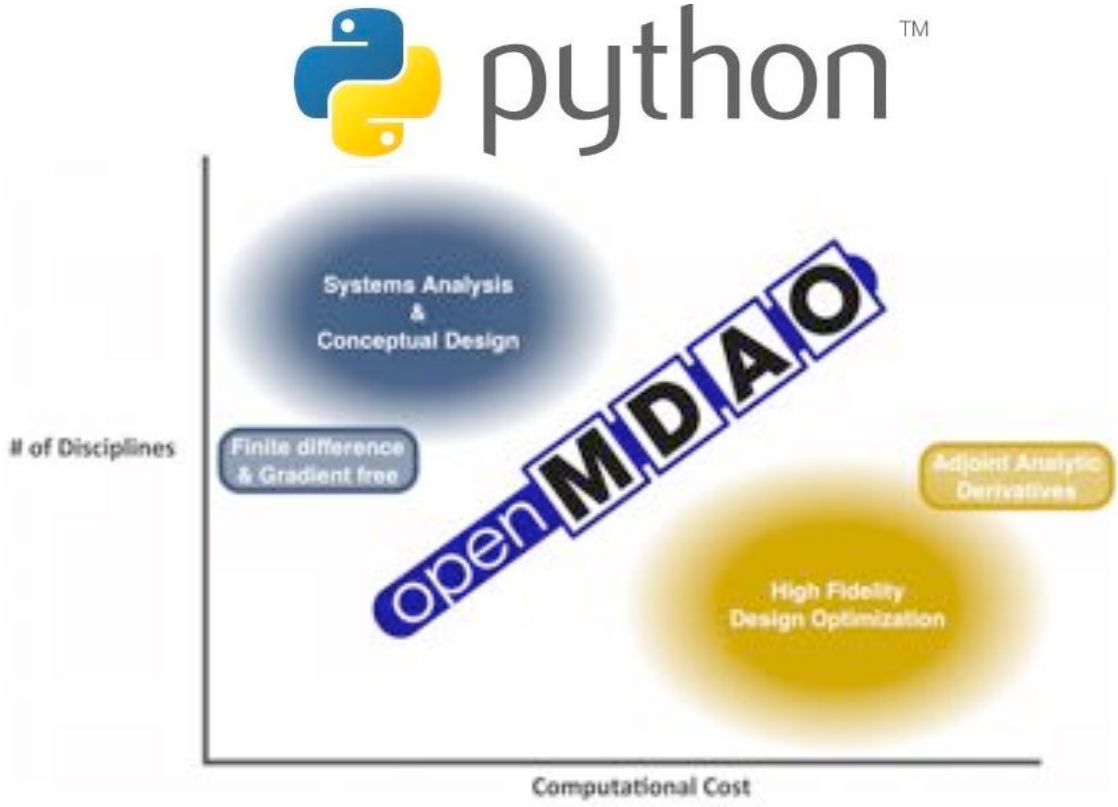


Software platform is built with Python using the OpenMDAO library

- **Most WISDEM modules are developed in Python using the OpenMDAO library**
  - Underlying analysis may be in C, C++ or Fortran

OpenMDAO (openmdao.org)

- **Open-source, python-based platform for systems analysis and multidisciplinary optimization**
- **Provides ”glue code” and “drivers/wrappers”**
- **Enables**
  - Model decomposition
  - Ease of development and maintenance
  - Tightly coupled solutions and parallel methods



WISDEM installation (essentials only)

- (Brief) Introduction to WISDEM and OpenMDAO
- **WISDEM installation (essentials only)**
- Tutorial 1: Run a simple WISDEM calculation with Jupyter Notebooks
- Some more detail on OpenMDAO
- Tutorial 2: Finding the Betz Limit through OpenMDAO optimization
- Tutorial 4: Modeling a whole turbine and plant

WISDEM install (essentials only): For full instructions see nwtc.nrel.gov/wisdem

- **Key steps**
  1. Download and install Anaconda3 64-bit from URL
  2. Setup new “conda environment” (provides digital sandbox to explore WISDEM without impacting any other part of your system)
  3. Install WISDEM and its dependencies
  4. Download WISDEM source code from GitHub

- **We want to install the code like a simple user but take a peek at the files like a developer**
- **Open the Anaconda Power Shell (Windows) or Terminal App (Mac) and do:**

```
conda config --add channels conda-forge
conda create -y --name wisdem-env python=3.7
conda activate wisdem-env
conda install –y wisdem git jupyter
git clone https://github.com/WISDEM/WISDEM.git
```

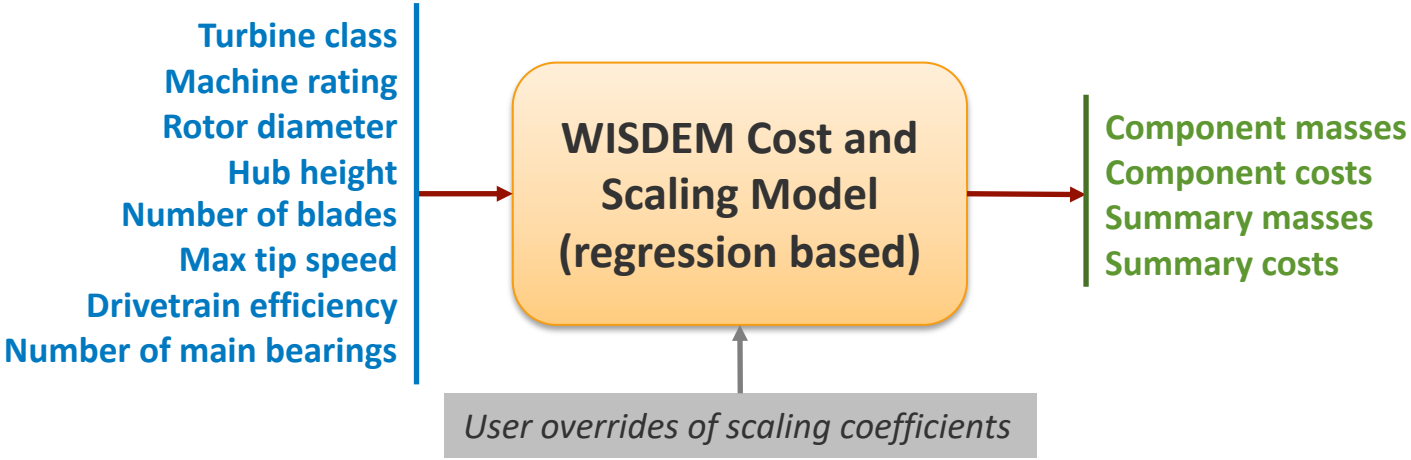
Tutorial 1: Run a simple WISDEM calculation with Jupyter Notebooks

- (Brief) Introduction to WISDEM and OpenMDAO
- WISDEM installation (essentials only)
- **Tutorial 1: Run a simple WISDEM calculation with Jupyter Notebooks**
- Some more detail on OpenMDAO
- Tutorial 2: Deriving the Betz Limit through OpenMDAO optimization
- Tutorial 4: Modeling a whole turbine and plant

Use WISDEM as a calculator to estimate component masses and costs from simple scaling relationships

- **WISDEM has multiple levels of fidelity, we will operate at the simplest level: "spreadsheet"-type calculation of component masses and cost**

- **We will: Populate inputs, execute model, list all the model inputs and outputs**
  - Will reveal some of the backend layers of OpenMDAO building blocks
  - Will ignore OpenMDAO syntax for now



Jupyter Notebook: Interacting with the Python “shell” through a browser in a live code “diary”

- **Jupyter Notebook is a web application that connects with your local python shell**

- **Allows for creating and sharing documents with**
  - Live code
  - Equations
  - Visualizations
  - Narrative text

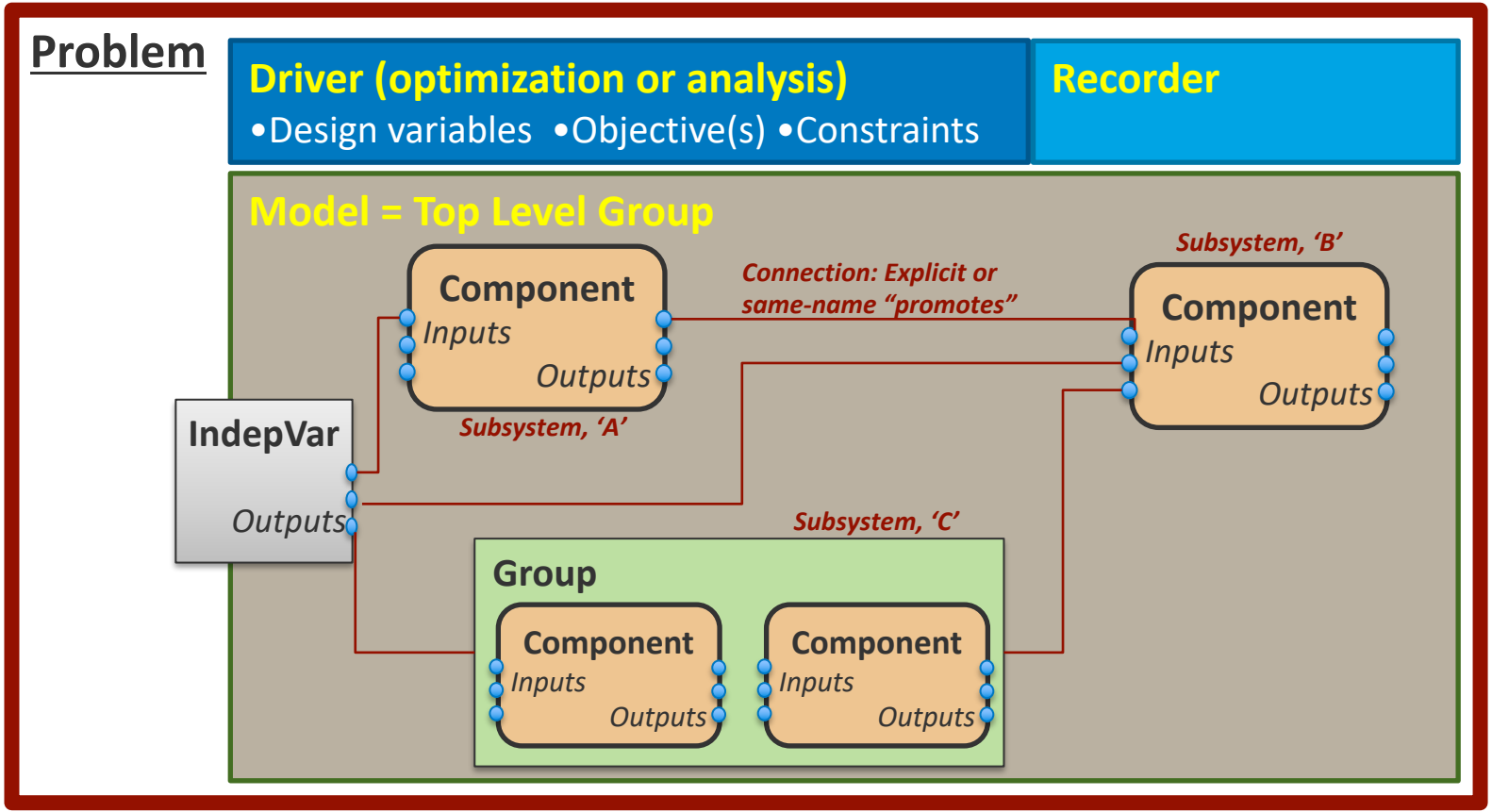
- **To get started with the WISDEM Jupyter Notebook tutorials we have to navigate to the right directory and start Jupyter**
- **Open the Anaconda Prompt (Windows) or Terminal App (Mac) and do:**

```
cd WISDEM/tutorial-notebooks
jupyter notebook
```

Some more detail on OpenMDAO

- (Brief) Introduction to WISDEM and OpenMDAO
- WISDEM installation (essentials only)
- Tutorial 1: Run a simple WISDEM calculation with Jupyter Notebooks
- **Some more detail on OpenMDAO**
- Tutorial 2: Deriving the Betz Limit through OpenMDAO optimization
- Tutorial 4: Modeling a whole turbine and plant

OpenMDAO building blocks and concepts



Tutorial 2: Finding the Betz Limit through OpenMDAO optimization

- (Brief) Introduction to WISDEM and OpenMDAO
- WISDEM installation (essentials only)
- Tutorial 1: Run a simple WISDEM calculation with Jupyter Notebooks
- Some more detail on OpenMDAO
- **Tutorial 2: Finding the Betz Limit through OpenMDAO optimization**
- Tutorial 4: Modeling a whole turbine and plant

OpenMDAO model building steps applied to the Betz Problem

Component Steps

- **Create an OpenMDAO *Component***
- **Add the actuator disk inputs and outputs**
- **Use *declare_partials()* to declare derivatives**
  - Finite difference or exact analytic options are available
- **Create a *compute()* method to compute outputs from inputs**
- **Create a *compute_partials()* method for the derivatives**

Group and Problem Steps

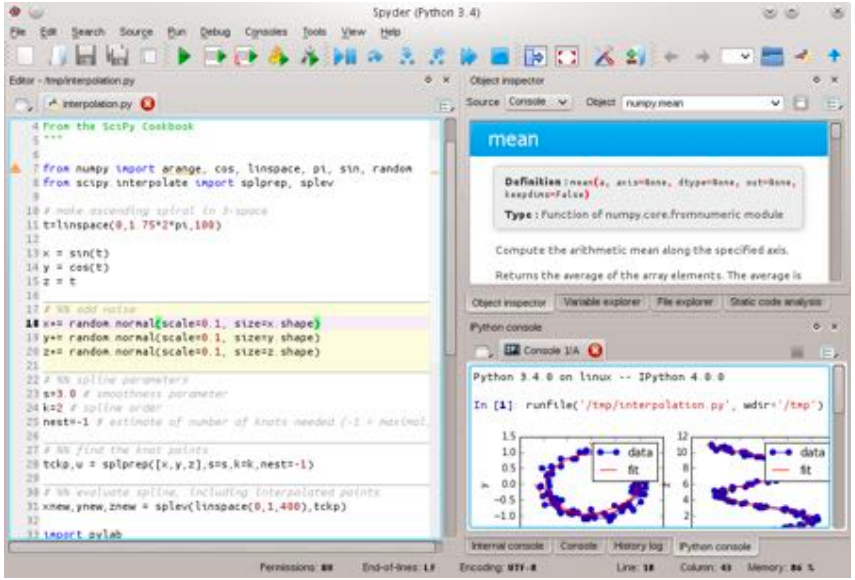
- **Create an OpenMDAO *Group*.**
  - Add a subsystem of independent variables
  - Add the disk Component as a subsystem
  - Connect variables through *connect()* statements or same name promotion
- **Create an OpenMDAO *Problem***
  - Set the model = Group instance
  - Add optimization *Driver*
  - Add design variables
  - Add the objective
- **Setup and run problem driver**

Tutorial 4: Modeling a whole turbine and plant

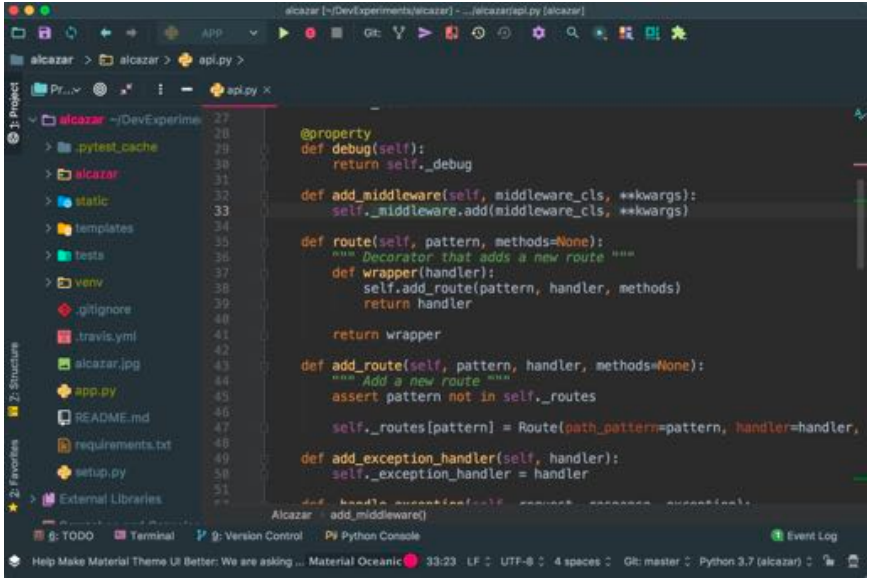
- (Brief) Introduction to WISDEM and OpenMDAO
- WISDEM installation (essentials only)
- Tutorial 1: Run a simple WISDEM calculation with Jupyter Notebooks
- Some more detail on OpenMDAO
- Tutorial 2: Finding the Betz Limit through OpenMDAO optimization
- **Tutorial 4: Modeling a whole turbine and plant**

There are many ways to run WISDEM (and python) beyond Jupyter Notebooks

Spyder for a Matlab-style Desktop



PyCharm or other IDE



Command line from Anaconda Prompt or Terminal App

```
[502 07:36 GBARTER-30696S:~/mdaoDevel/WISDEM/wisdem/assemblies/land_based $python land_based_noGenerator_noBOS_lcoe.py
 Running initialization: ../../rotorse/turbine_inputs/nrel5mw_mod_update.yaml
 Complete: Load Input File:      0.001048 s
 Complete: Geometry Analysis:    0.565820 s
```