1.1 Installing SUAVE

Basic and Developer Installation Guide

1.2 SUAVE Code Architecture

OVERVIEW

SUAVE

- SUAVE
  - trunk/SUAVE
    - Analyses
    - Attributes
    - Components
    - Core
    - Input_Output
    - Methods
    - Optimization
    - Plots
    - Plugins
    - Surrogate
  - regressions
    - scripts

SUAVE

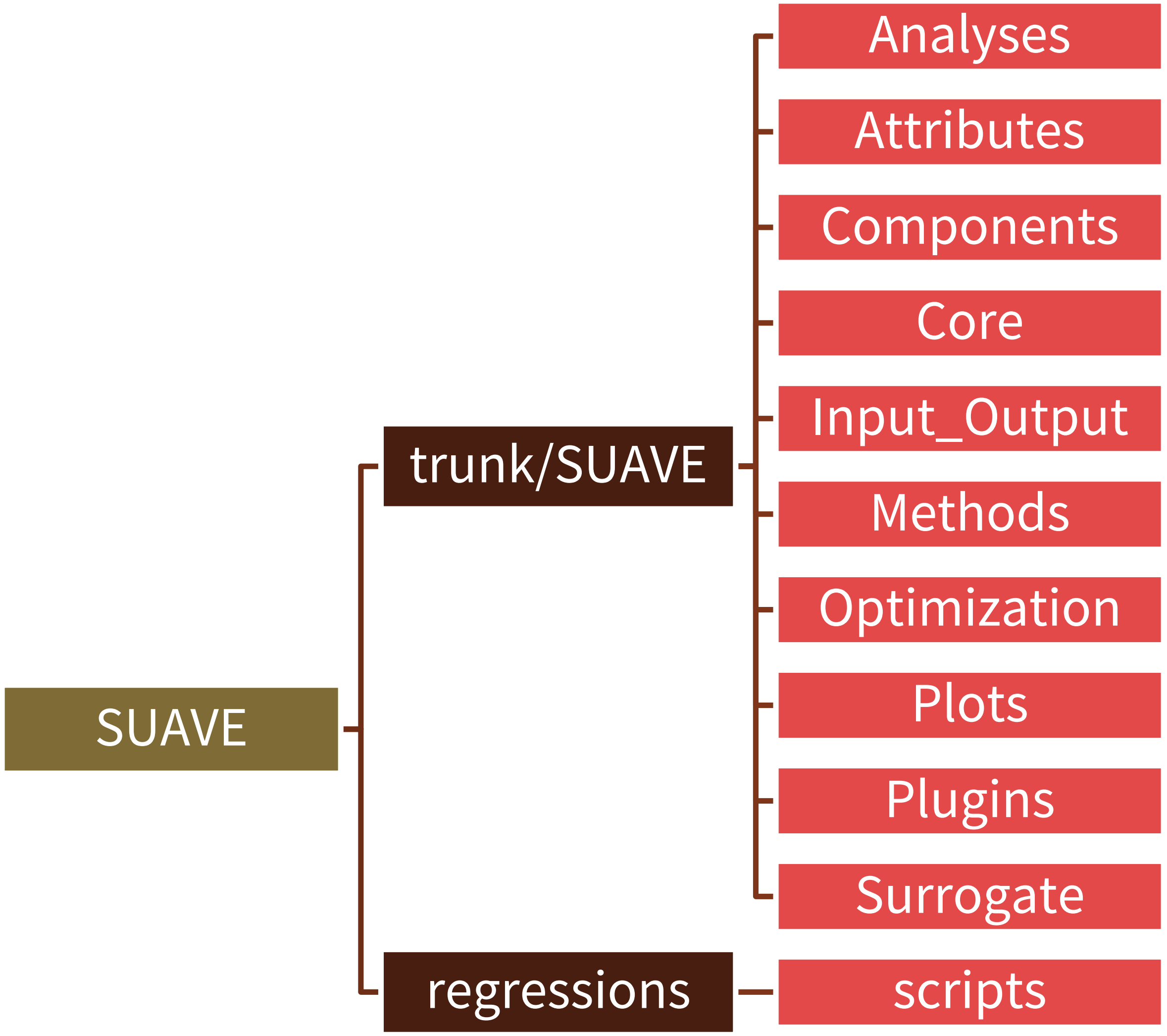
SUAVE CODE

- SUAVE
  - trunk/SUAVE
    - Analyses
    - Attributes
    - Components
    - Core
    - Input_Output
    - Methods
    - Optimization
    - Plots
    - Plugins
    - Surrogate
  - regressions
    - scripts

Function calls to initialize various disciplines

SUAVE CODE

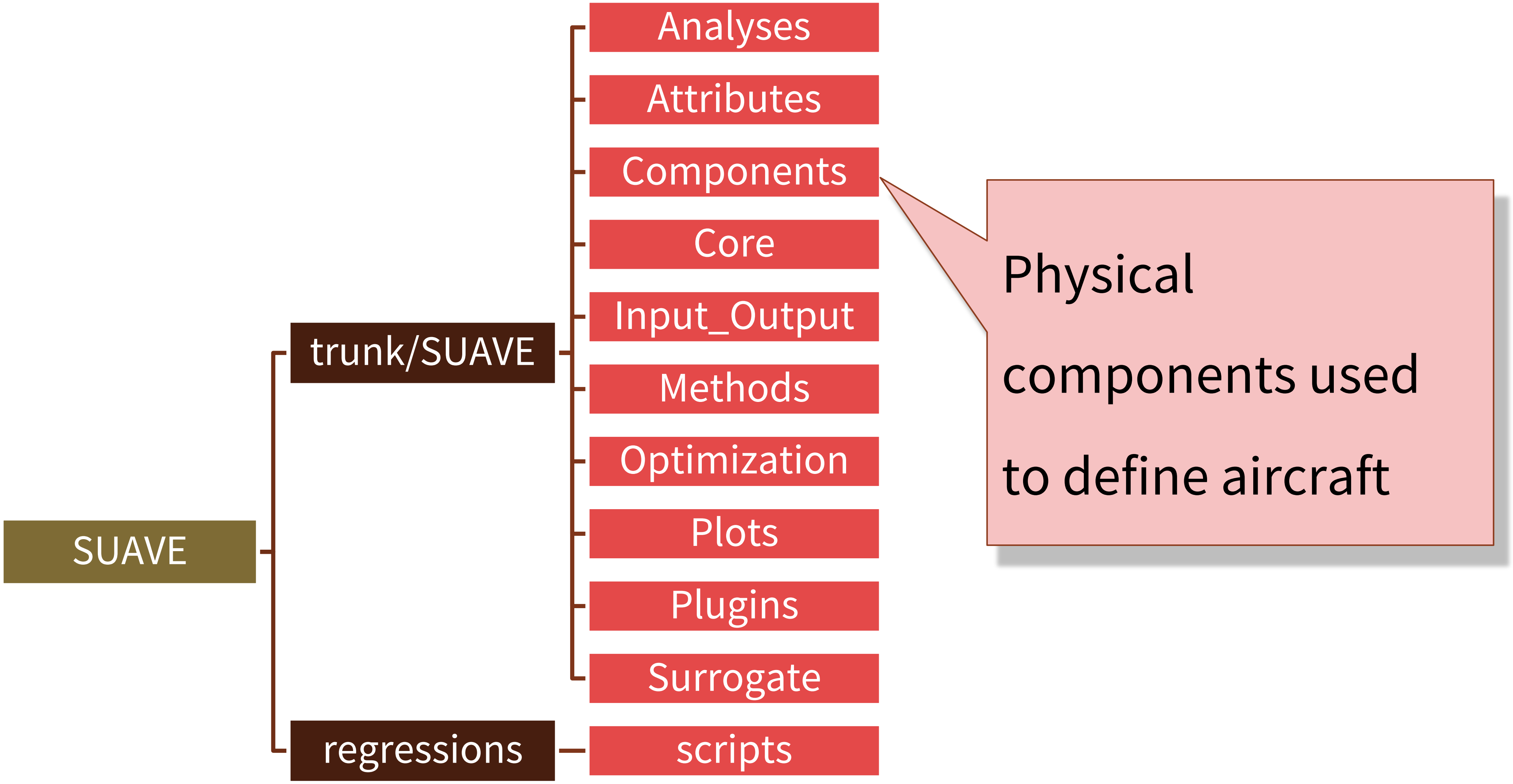
SUAVE



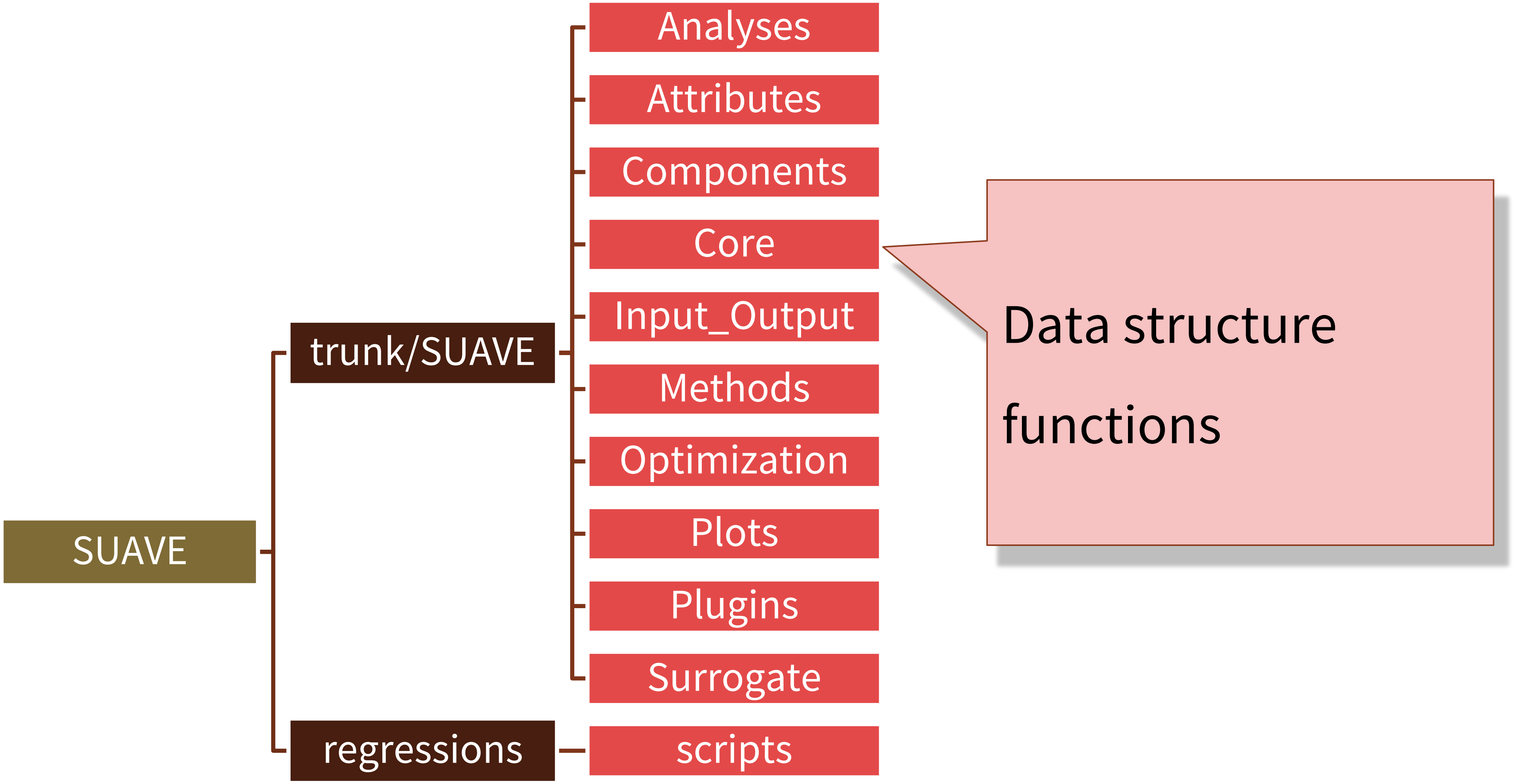
Properties of fluids and atmosphere

SUAVE CODE

SUAVE

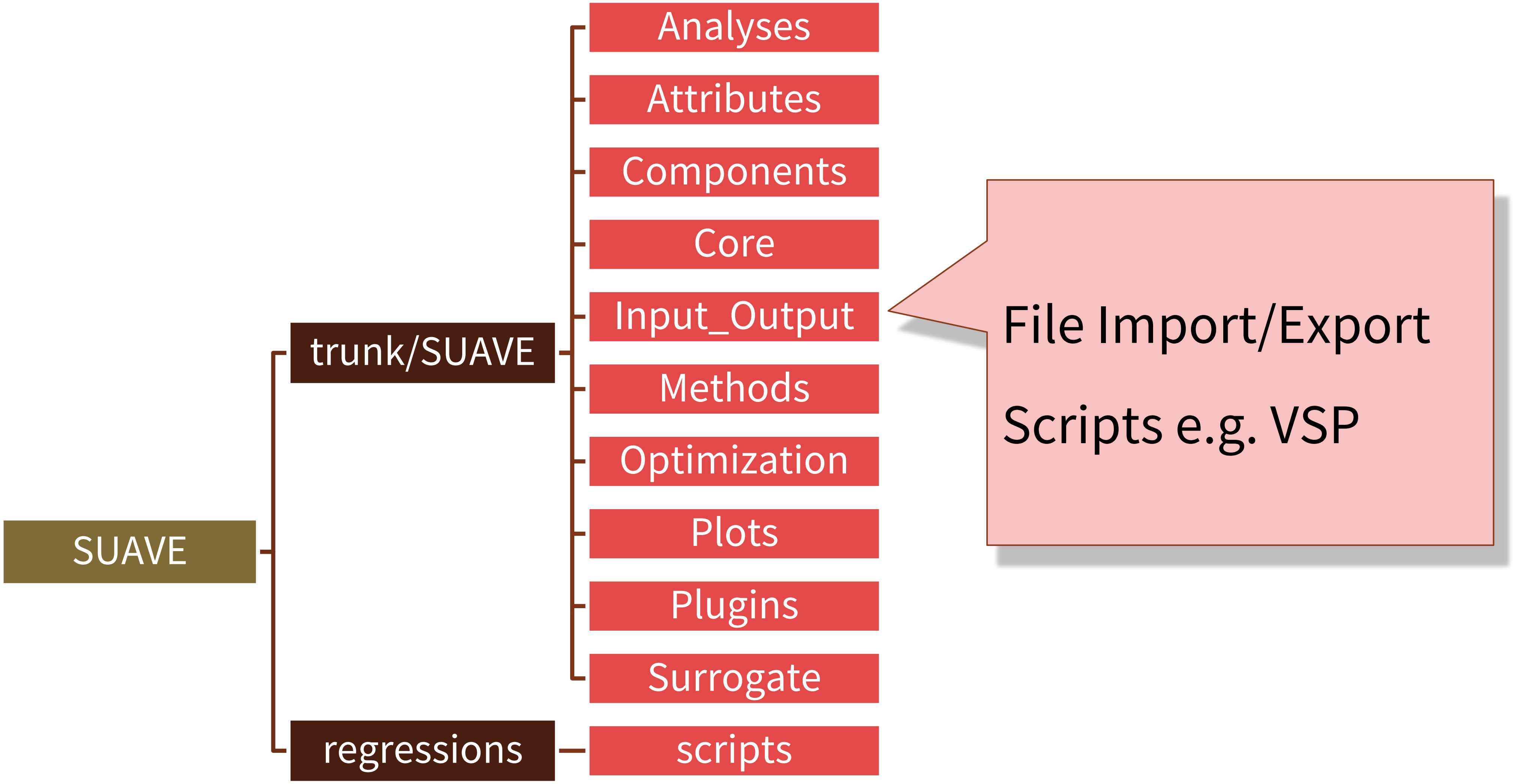


SUAVE CODE


SUAVE
trunk/SUAVE
regressions
Analyses
Attributes
Components
Core
Input_Output
Methods
Optimization
Plots
Plugins
Surrogate
scripts
Data structure
functions

SUAVE CODE

SUAVE



SUAVE CODE

SUAVE

- SUAVE
  - trunk/SUAVE
    - Analyses
    - Attributes
    - Components
    - Core
    - Input_Output
    - Methods
    - Optimization
    - Plots
    - Plugins
    - Surrogate
  - regressions
    - scripts

Functions called by Analyses scripts

SUAVE CODE

- SUAVE
  - trunk/SUAVE
    - Analyses
    - Attributes
    - Components
    - Core
    - Input_Output
    - Methods
    - Optimization
    - Plots
    - Plugins
    - Surrogate
  - regressions
    - scripts

Optimization wrappers & framework

SUAVE CODE

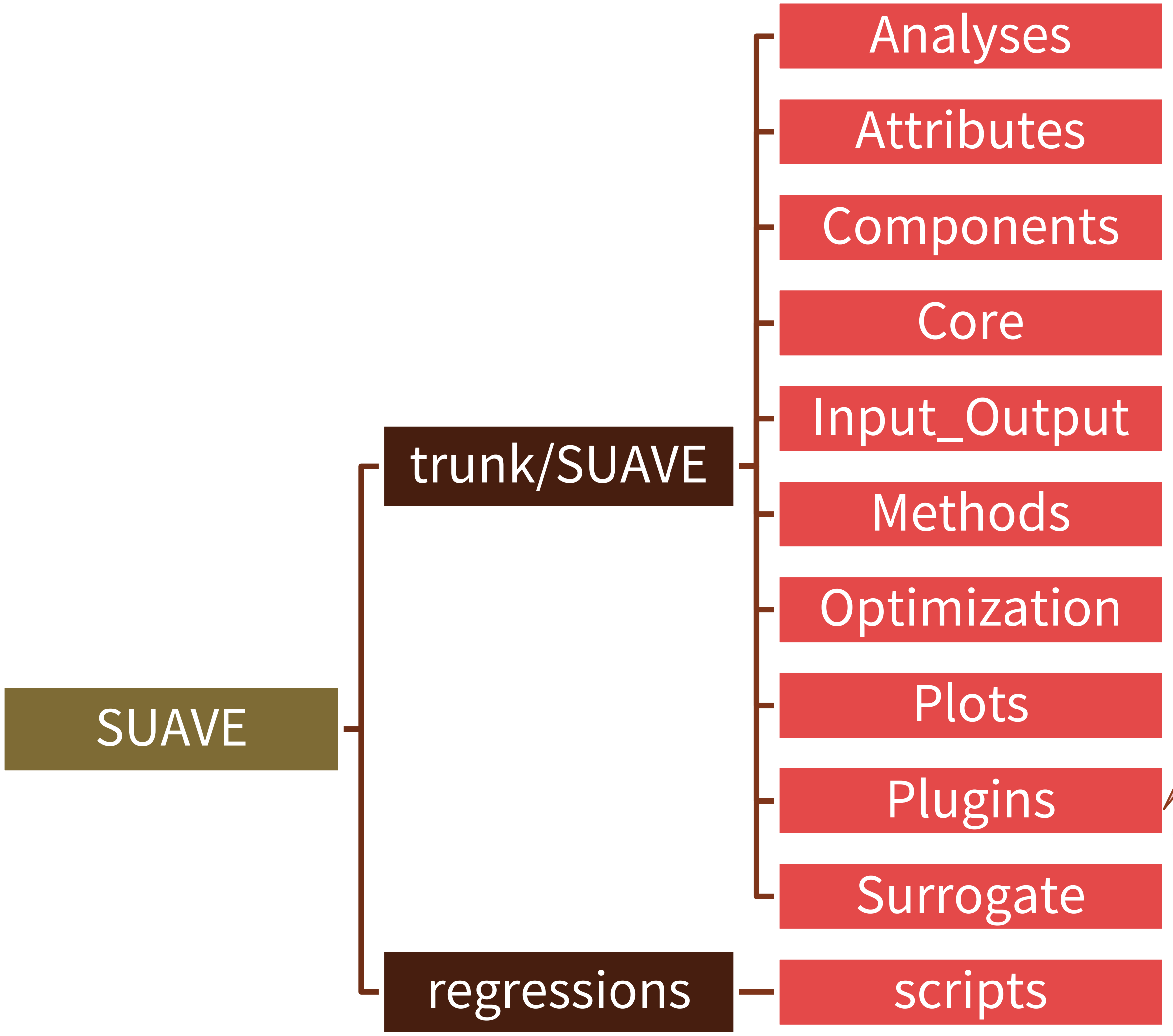
SUAVE

- SUAVE
  - trunk/SUAVE
    - Analyses
    - Attributes
    - Components
    - Core
    - Input_Output
    - Methods
    - Optimization
    - Plots
    - Plugins
    - Surrogate
  - regressions
    - scripts

Plotting functions

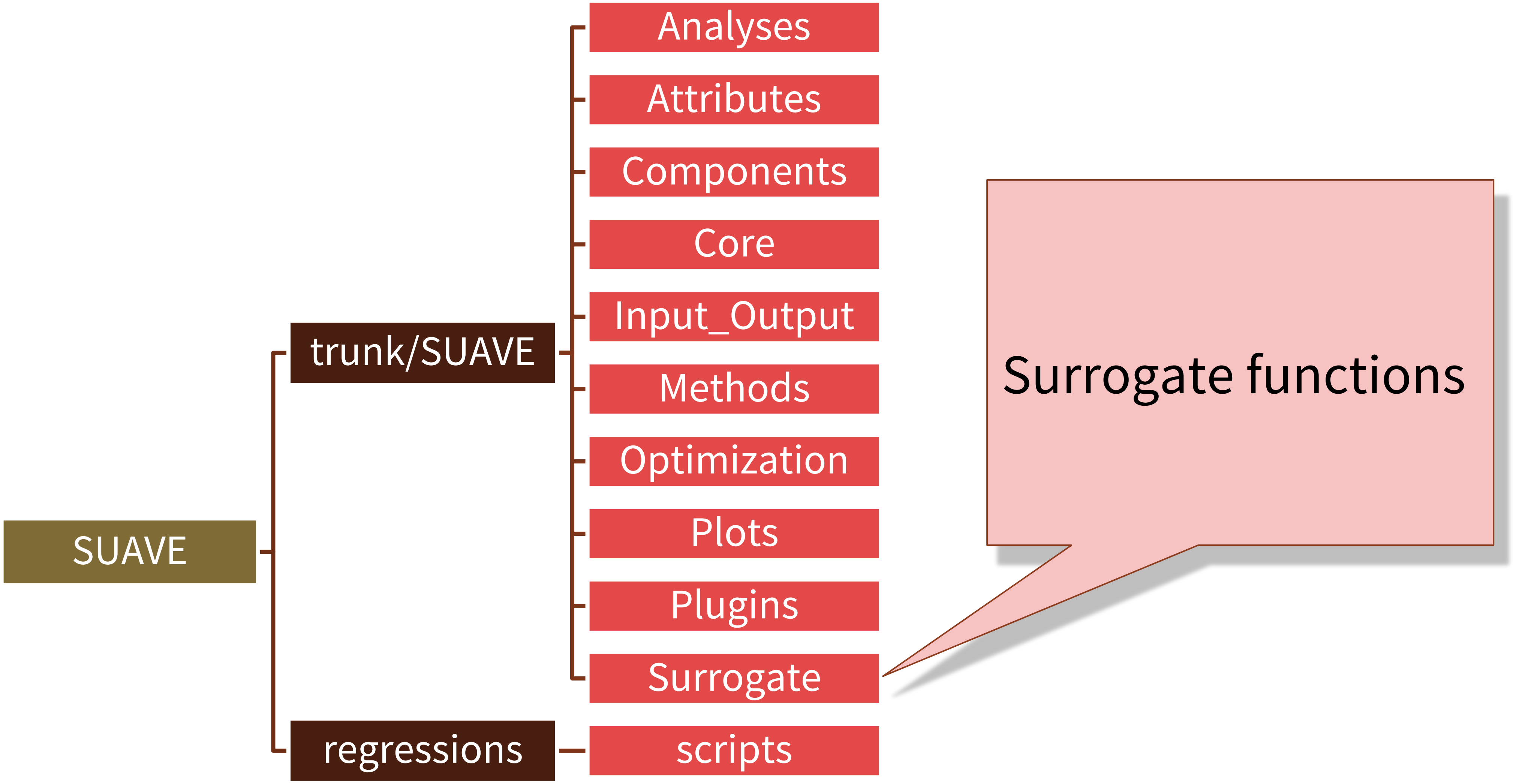
SUAVE CODE

SUAVE

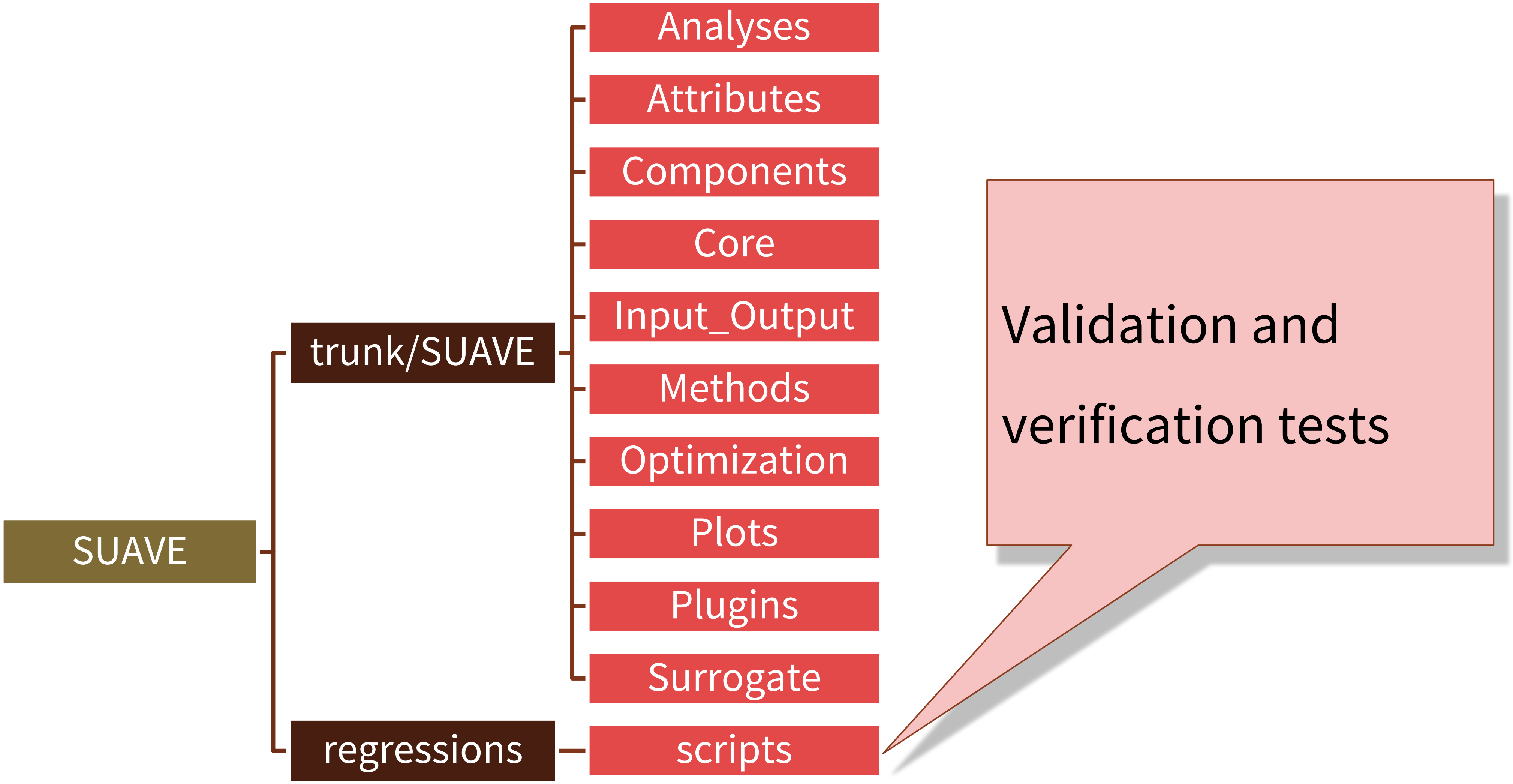


Supplemental units and measurement functions

SUAVE CODE


SUAVE
trunk/SUAVE
regressions
Analyses
Attributes
Components
Core
Input_Output
Methods
Optimization
Plots
Plugins
Surrogate
scripts
Surrogate functions

SUAVE CODE


SUAVE
trunk/SUAVE
regressions
Analyses
Attributes
Components
Core
Input_Output
Methods
Optimization
Plots
Plugins
Surrogate
scripts
Validation and verification tests

2.1 SUAVE Code Architecture

SUAVE/trunk/SUAVE/Analyses

SUAVE/trunk/SUAVE

SUAVE

- Analyses
- Attributes
- Components
- Core
- Input_Output
- Methods
- Optimization
- Plots
- Plugins
- Surrogates

SUAVE/trunk/SUAVE/Analyses

- SUAVE
  - Analyses
    - Aerodynamics
    - Atmospheric
    - Costs
    - Energy
    - Mission
    - Noise
    - Planets
    - Sizing
    - Stability
    - Weights
  - Attributes
  - Components
  - Core
  - Input_Output
  - Methods
  - Optimization
  - Plots
  - Plugins
  - Surrogates

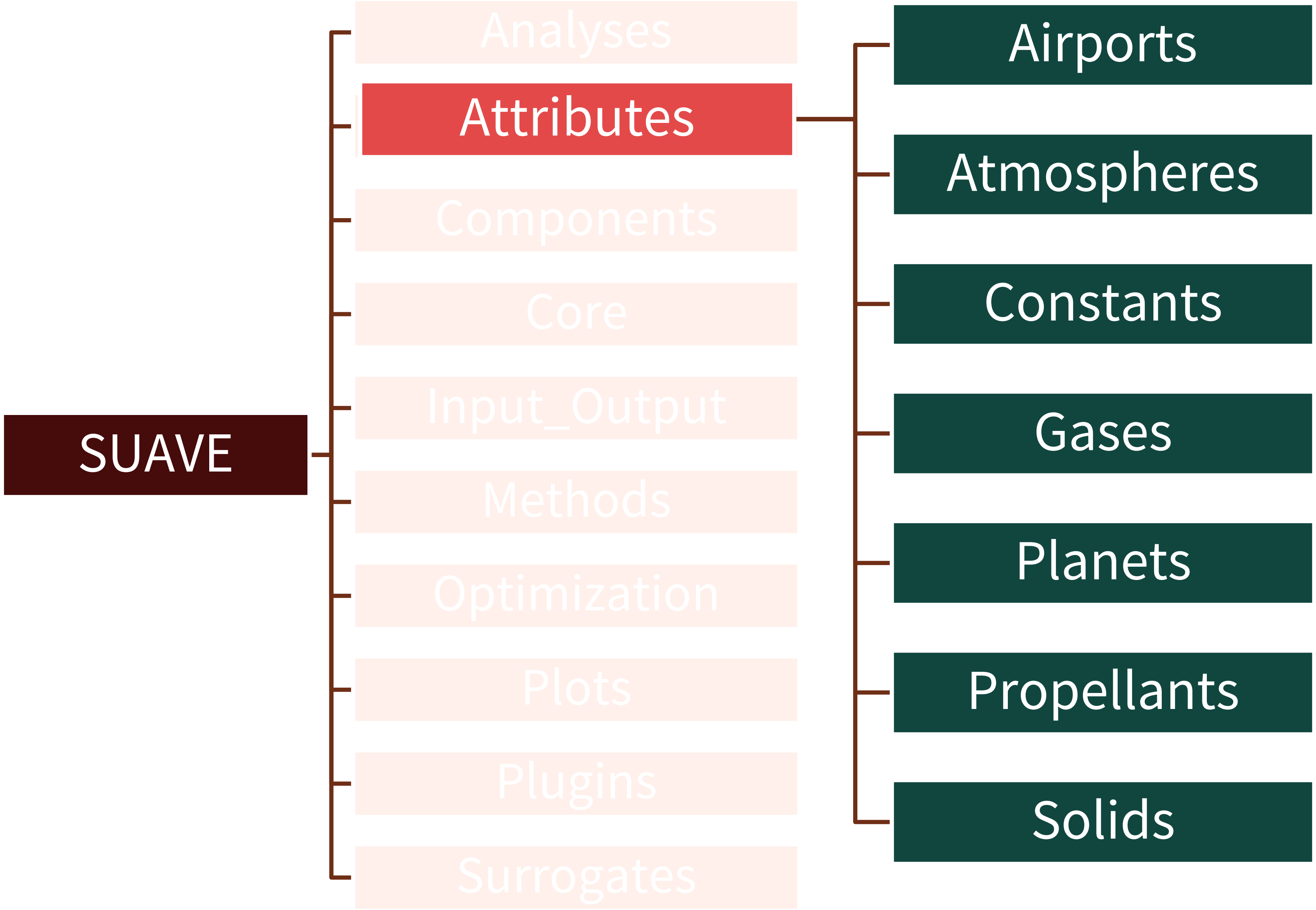
2.2 SUAVE Code Architecture

SUAVE/trunk/SUAVE/Attributes

SUAVE/trunk/SUAVE

- SUAVE
  - Analyses
  - Attributes
  - Components
  - Core
  - Input_Output
  - Methods
  - Optimization
  - Plots
  - Plugins
  - Surrogates

SUAVE/trunk/SUAVE/Attributes



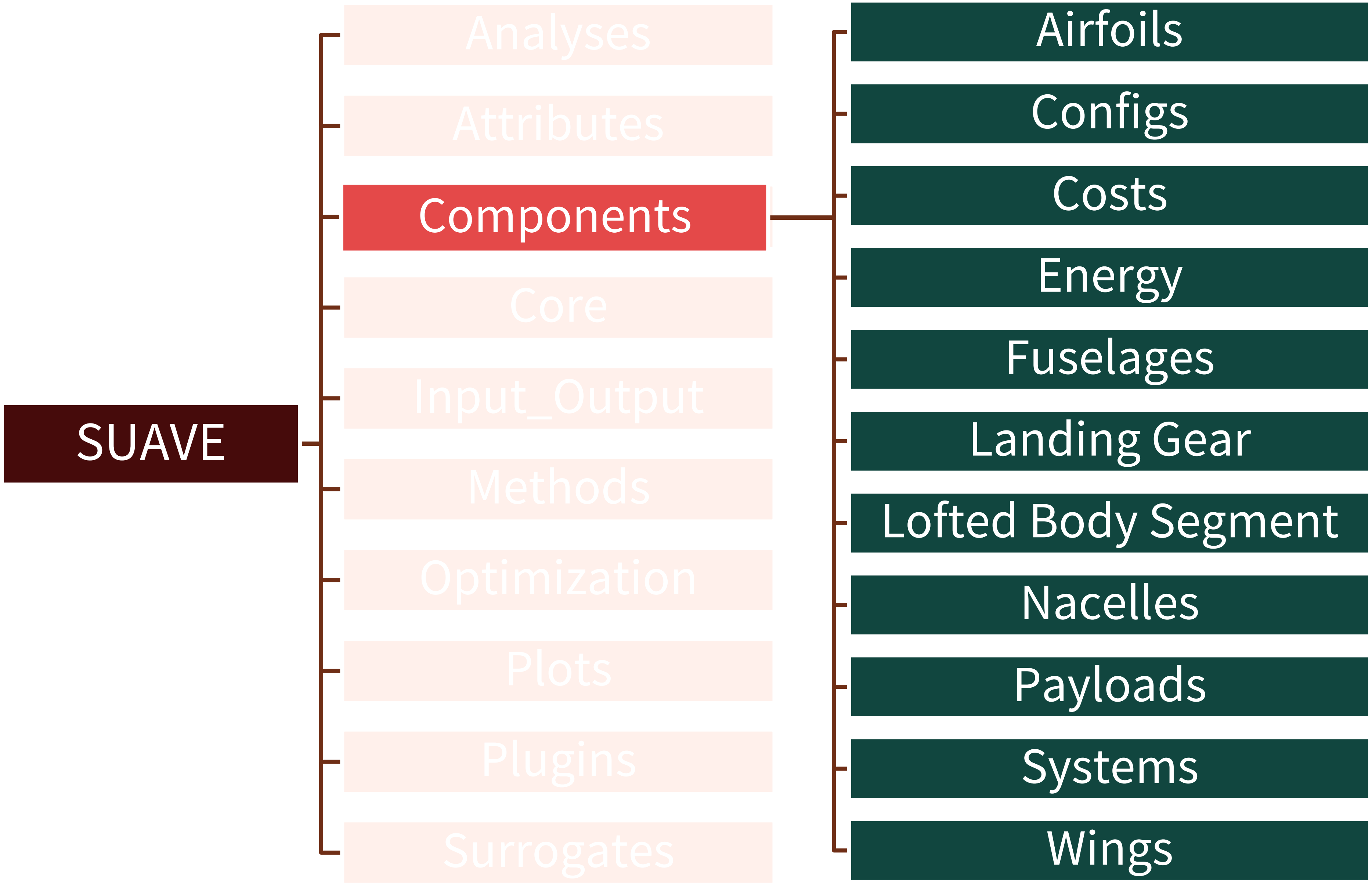
2.3 SUAVE Code Architecture

SUAVE/trunk/SUAVE/Components

SUAVE/trunk/SUAVE

- SUAVE
  - Analyses
  - Attributes
  - Components
  - Core
  - Input_Output
  - Methods
  - Optimization
  - Plots
  - Plugins
  - Surrogates

SUAVE/trunk/SUAVE/Components



SUAVE/trunk/SUAVE/Components

SUAVE

- Analyses
- Attributes
- Components
  - Airfoils
  - Configs
  - Costs
  - Energy
    - Charging
    - Converters
    - Distributors
    - Networks
    - Peripherals
    - Processes
    - Storages
  - Fuselages
  - Landing Gear
  - Lofted Body Segment
  - Nacelles
  - Payloads
  - Systems
  - Wings
- Core
- Input_Output
- Methods
- Optimization
- Plots

2.4 SUAVE Code Architecture

SUAVE/trunk/SUAVE/Input_Output

SUAVE/trunk/SUAVE

- SUAVE
  - Analyses
  - Attributes
  - Components
  - Core
  - Input_Output
  - Methods
  - Optimization
  - Plots
  - Plugins
  - Surrogates

SUAVE/trunk/SUAVE/Input_Output

- SUAVE
  - Analyses
  - Attributes
  - Components
  - Core
  - Input_Output
    - D3JS
    - GMSH
    - OpenVSP
    - Results
    - SU2
    - VTK
    - SML
  - Methods
  - Optimization
  - Plots
  - Plugins
  - Surrogates

2.5 SUAVE Code Architecture

SUAVE/trunk/SUAVE/Methods

SUAVE/trunk/SUAVE

- SUAVE
  - Analyses
  - Attributes
  - Components
  - Core
  - Input_Output
  - Methods
  - Optimization
  - Plots
  - Plugins
  - Surrogates

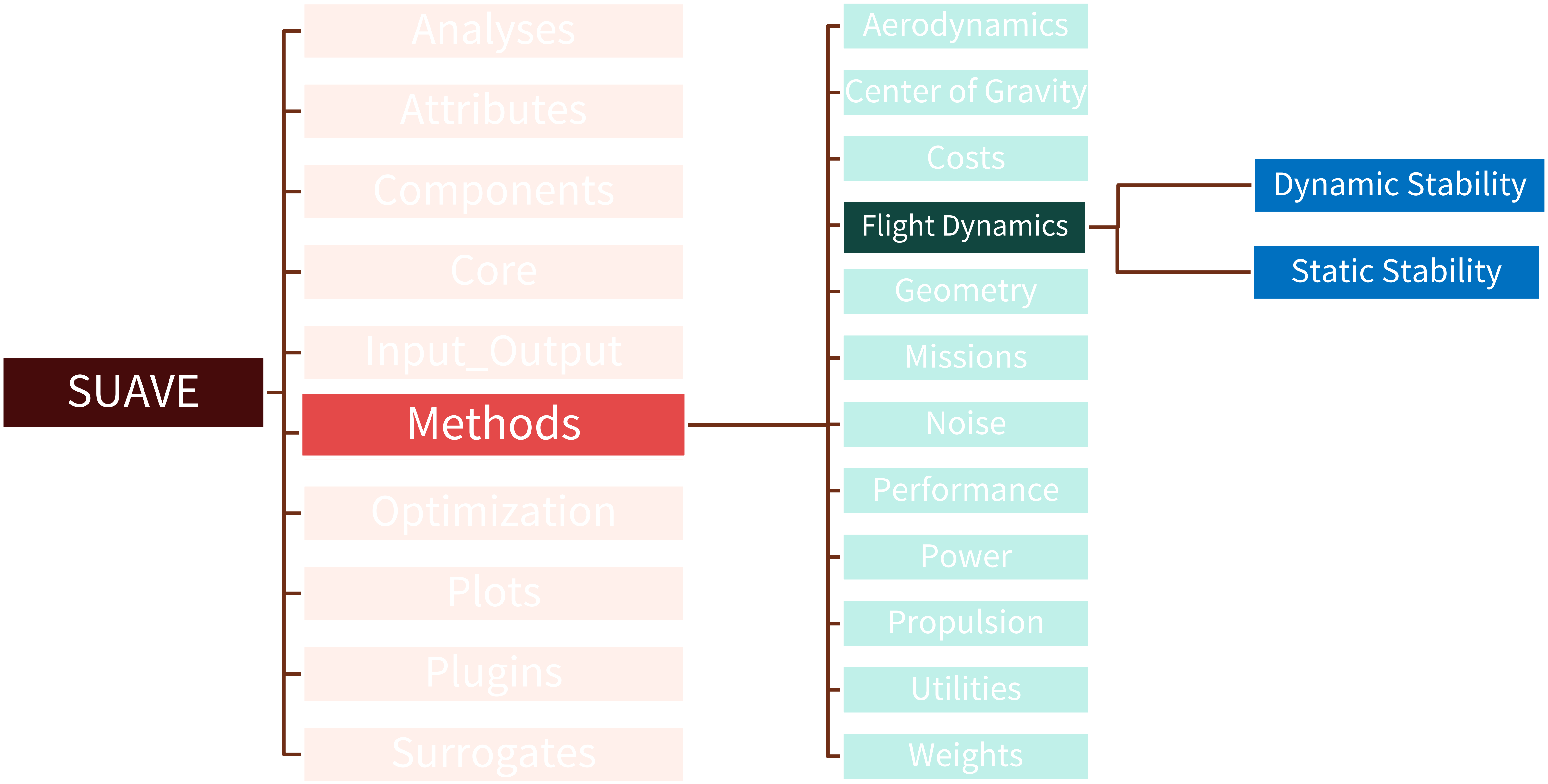
SUAVE/trunk/SUAVE/Methods

- SUAVE
  - Analyses
  - Attributes
  - Components
  - Core
  - Input_Output
  - **Methods**
    - Aerodynamics
    - Center of Gravity
    - Costs
    - Flight Dynamics
    - Geometry
    - Missions
    - Noise
    - Performance
    - Power
    - Propulsion
    - Utilities
    - Weights
  - Optimization
  - Plots
  - Plugins
  - Surrogates

SUAVE/trunk/SUAVE/Methods

- SUAVE
  - Analyses
  - Attributes
  - Components
  - Core
  - Input_Output
  - Methods
    - Aerodynamics
      - AERODAS
      - Airfol_Panel_Method
      - AVL
      - Common
      - Fidelity Zero
      - Lifting_Line
      - OpenVSP_Wave_Drag
      - Supersonic_Zero
    - Center of Gravity
    - Costs
    - Flight Dynamics
    - Geometry
    - Missions
    - Noise
    - Performance
    - Power
    - Propulsion
    - Utilities
    - Weights
  - Optimization
  - Plots
  - Plugins
  - Surrogates

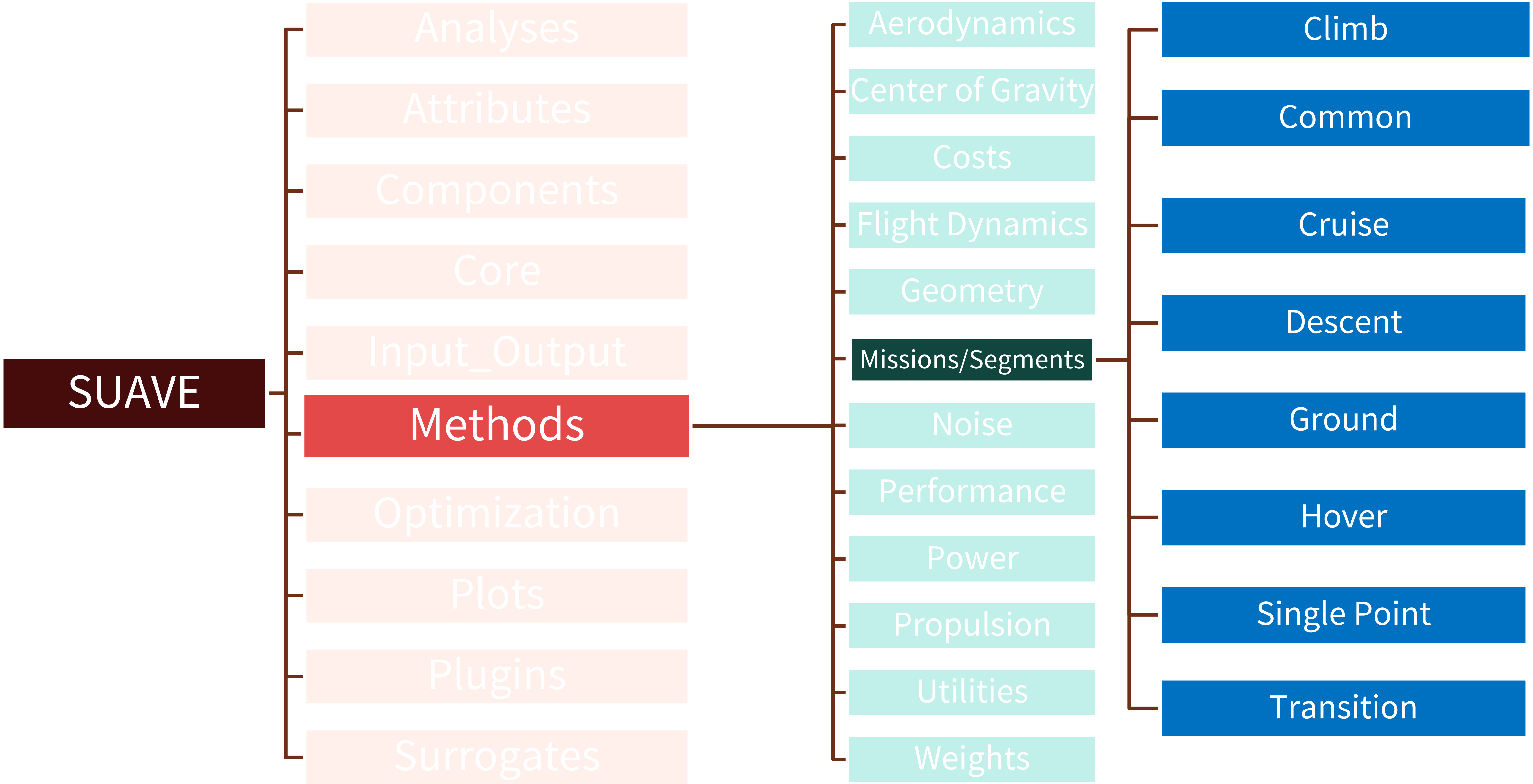
SUAVE/trunk/SUAVE/Methods



SUAVE/trunk/SUAVE/Methods

- SUAVE
  - Analyses
  - Attributes
  - Components
  - Core
  - Input_Output
  - Methods
    - Aerodynamics
    - Center of Gravity
    - Costs
    - Flight Dynamics
    - Geometry
      - Two Dimensional
      - Three Dimensional
    - Missions
    - Noise
    - Performance
    - Power
    - Propulsion
    - Utilities
    - Weights
  - Optimization
  - Plots
  - Plugins
  - Surrogates

SUAVE/trunk/SUAVE/Methods



SUAVE/trunk/SUAVE/Methods

- SUAVE
  - Analyses
  - Attributes
  - Components
  - Core
  - Input_Output
  - Methods
    - Aerodynamics
    - Center of Gravity
    - Costs
    - Flight Dynamics
    - Geometry
    - Missions
    - Noise
      - Boom
      - Certification
      - Fidelity Zero
      - Fidelity One
    - Performance
    - Power
    - Propulsion
    - Utilities
    - Weights
  - Optimization
  - Plots
  - Plugins
  - Surrogates

SUAVE/trunk/SUAVE/Methods

SUAVE

- Analyses
- Attributes
- Components
- Core
- Input_Output
- Methods
  - Aerodynamics
  - Center of Gravity
  - Costs
  - Flight Dynamics
  - Geometry
  - Missions
  - Noise
  - Performance
  - Power
  - Propulsion
  - Utilities
  - Weights
    - Buildups
    - Correlations
- Optimization
- Plots
- Plugins
- Surrogates

SUAVE

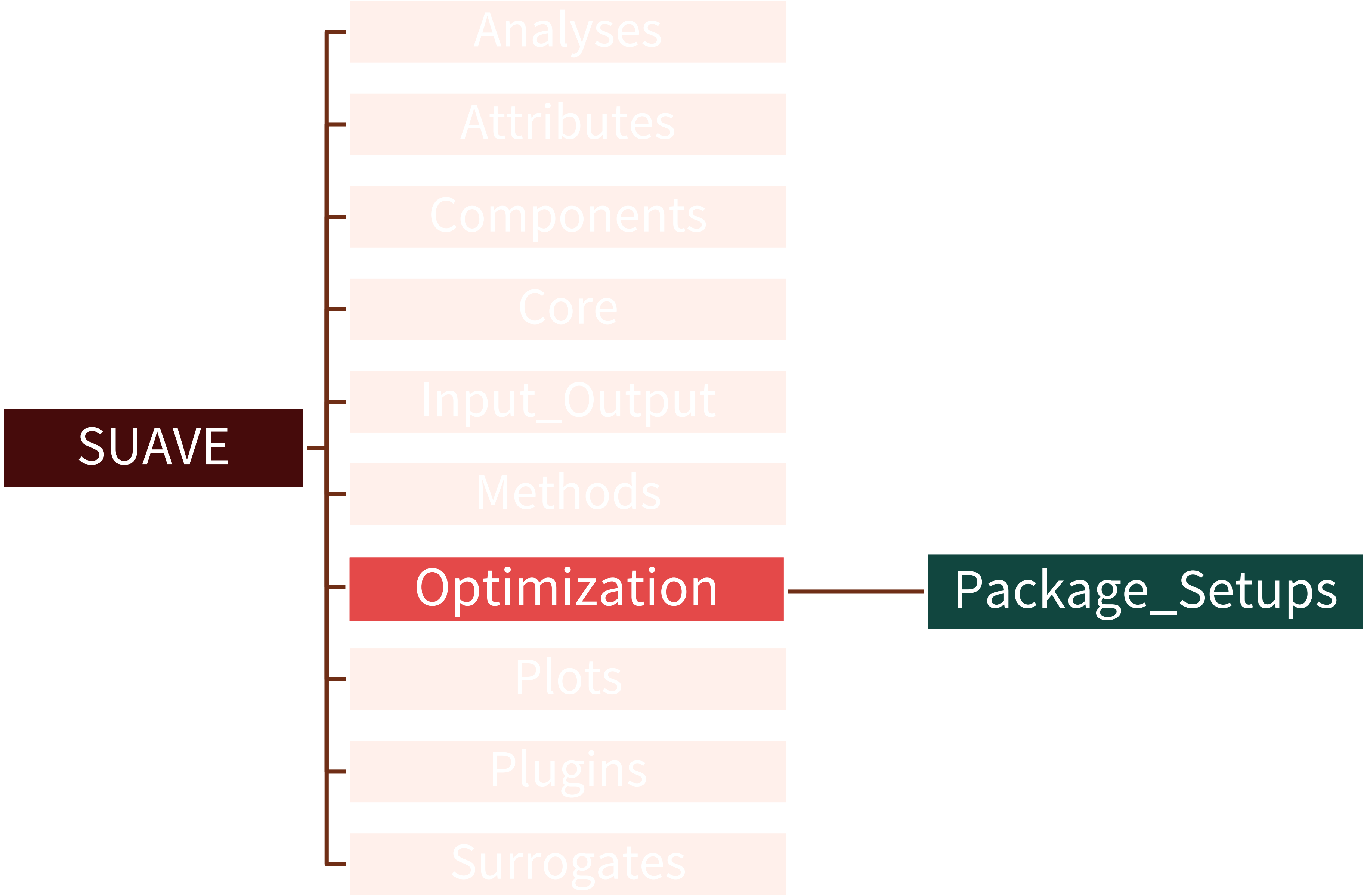
2.6 SUAVE Code Architecture

SUAVE/trunk/SUAVE/Optimization

SUAVE/trunk/SUAVE

- SUAVE
  - Analyses
  - Attributes
  - Components
  - Core
  - Input_Output
  - Methods
  - Optimization
  - Plots
  - Plugins
  - Surrogates

SUAVE/trunk/SUAVE



2.7 SUAVE Code Architecture

SUAVE/trunk/SUAVE/Plots

SUAVE/trunk/SUAVE

- SUAVE
  - Analyses
  - Attributes
  - Components
  - Core
  - Input_Output
  - Methods
  - Optimization
  - Plots
  - Plugins
  - Surrogates

SUAVE/trunk/SUAVE

- SUAVE
  - Analyses
  - Attributes
  - Components
  - Core
  - Input_Output
  - Methods
  - Optimization
  - Plots
    - Geometry
    - Performance
  - Plugins
  - Surrogates

3.1 SUAVE Tutorials

Boeing 737 Mission Analysis Using Fidelity Zero (SUAVE's VLM)

Overview of tut_mission_B737.py

- This tutorial shows how the user can build a conventional aircraft and analyze its performance over the course of a mission. The focus is on providing general information that can apply to setup for a variety of vehicles.

Objectives of Tutorial

SUAVE

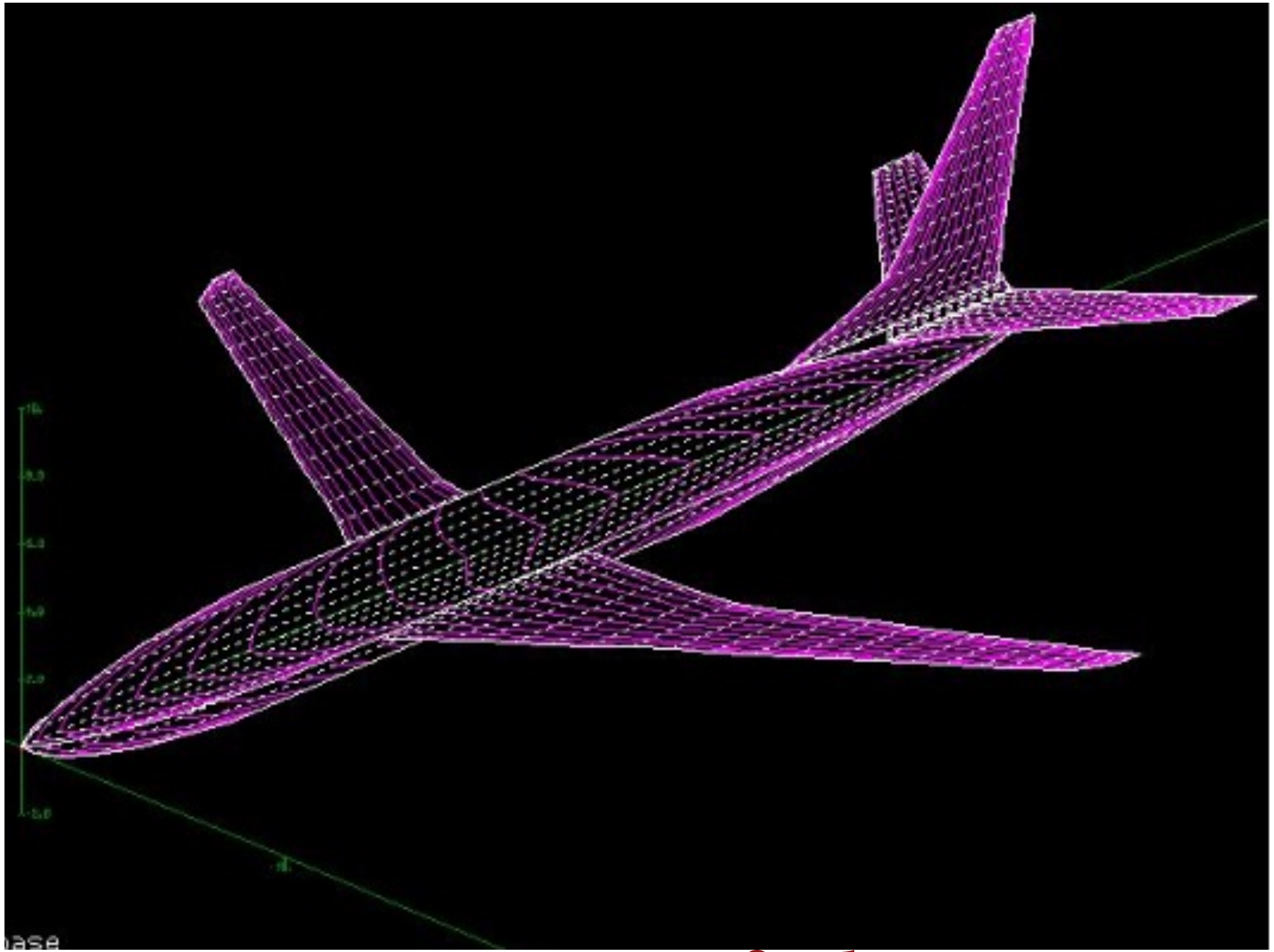
- Understand the basic functions of a SUAVE mission script
- Define geometry aircraft
  - Turbofan modeling tutorial is covered here
- Assign various computational methods to analyze a mid-range transport aircraft
- Write a detailed flight profile, including takeoff and landing
- Analyze results

3.2 SUAVE Tutorials

Boeing 737 Mission Analysis Using AVL

Overview of tut_Mission_AVL_Tutorial_B737

- This tutorial shows how the user can use Athena Vortex Lattice (AVL) developed by MIT within the SUAVE framework to study analyze the performance of a Boeing 737-800 over the course of a mission.



SUAVE

Objectives of Tutorial

- Understand the basic functions in a SUAVE mission script
- Highlight the areas of the code that differ from tut_mission_B737.py tutorial
- Understand how to use specific vehicle configurations specific flight segments in a flight profile
- Understand how to change analysis techniques and methodologies i.e. aerodynamics, weights, stability
- Analyze results

3.3 SUAVE Tutorials

Regional Jet Fuel Burn Optimization

Overview of tut_Regional_Jet_Optimization

SUAVE

- This tutorial seeks to provide an example of the more general optimization capabilities in SUAVE and provides more information on how to modify the code for different uses.
- Specifically, the optimization of the planform of a regional jet (Embraer-190) to minimize fuel burn will be covered.

Overview of tut_Regional_Jet_Optimization

Important Files

- Optimize.py - Defines the optimization framework of the problem, wherein one minimizes an assigned objective, subject to certain constraints, by altering some design variables.
- Procedure.py - This links everything together by defining the steps you would use to size and analyze the aircraft at each optimizer iteration.
- Vehicle.py - Initializes the vehicle (or vehicles if desired) used in the optimization problem.
- Analyses.py - Defines the set of features that are used in this particular problem (e.g. weights correlations, aerodynamics correlations, etc.).
- Missions.py - Initializes the missions that are run at each iteration. In this case, only a single mission is run.
- Plot_Mission.py - Plots the mission outputs.

Objectives of Tutorial

- Understand the general framework for carrying out optimizations in SUAVE
- Have familiarity with Nexus, SUAVE's data object used to communicate with external optimizers
- Modify optimization scripts (design variables, constraints, objective functions) and post processing functions
- Modify optimizers including the tolerances, step-size and sampling size

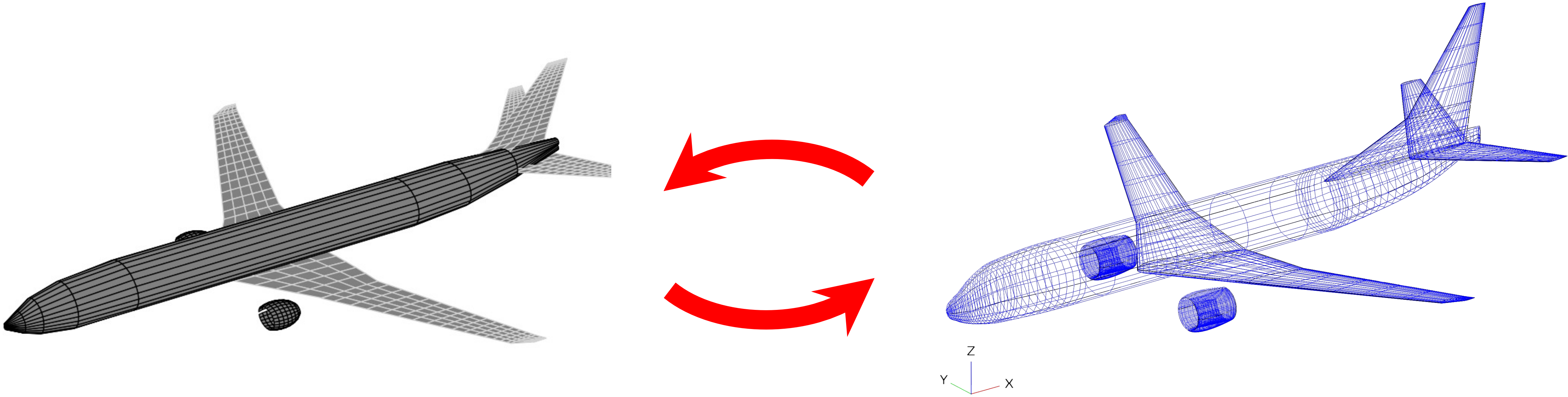
3.4 SUAVE Tutorials

Importing and Exporting aircraft to OpenVSP

Overview of tut_VSP_Import_Export_B737.py

SUAVE

- This tutorial covers how to utilize some of SUAVE's importing and exporting functions to generate and read geometry into OpenVSP for external analysis



SUAVE **OpenVSP**

Objectives to Tutorial

- Understand how to import and export geometry from OpenVSP
- Understand what additional information is required after importing a geometry to run a mission i.e. appending a network, vehicle mass
- Understand the limitations of the geometry parametrization function in SUAVE

3.5 SUAVE Tutorials

Aircraft Single point analysis

Overview of tut_Single_Point_Analysis_X57Mod2_AVL.py

- This tutorial details how to set up a single-point analysis using AVL for an aircraft simulate performance in any flight condition.
- The NASA X57 Maxwell Modification II is used in this example

Objectives to Tutorial

- Learn how to set up a single point analysis
- Utilize AVL to perform trimmed and untrimmed studies

3.6 SUAVE Tutorials

Aircraft Flight Polar Alayses

SUAVE

Overview of tut_Aerodynamic_Polars_X57Mod3.py

- This tutorial covers how to generate the coefficients of lift, drag and moment as a function of angle of attack for an aircraft.
- The NASA X57 Maxwell Modification III is used in this example.

Objectives to Tutorial

- Learn how to set up angle of attack analyses
- Revisit how to change the aerodynamic/stability analysis routines

3.8 SUAVE Tutorials

Control Surface Sizing

Overview of tut_Control_Surface_Sizing_Navion.py

- This tutorial covers the sizing optimization of control surfaces for the Navion Aircraft using SUAVE's Optimization Wrapper.
- Both SUAVE's VLM and AVL are used in analysis the framework.

Objectives of Tutorial

- Starting with aircraft with no control surfaces, develop a sizing routine to appropriate size flaps, elevators, ailerons and rudders to meet specific handling criteria
- Understand the basic stability and trim requirements for aircraft, where to find handling criteria and how to express them as mathematical formulations within an optimization
- Review the structure of SUAVE's optimization framework
- Define desired objective functions, design variables, contains, and sizing functions
- Define and execute single point analyses
- Utilize AVL to perform trimmed and untrimmed studies

Navion Geometry in SUAVE

Stick-Fixed Stability and Aerodynamics Optimization Setup

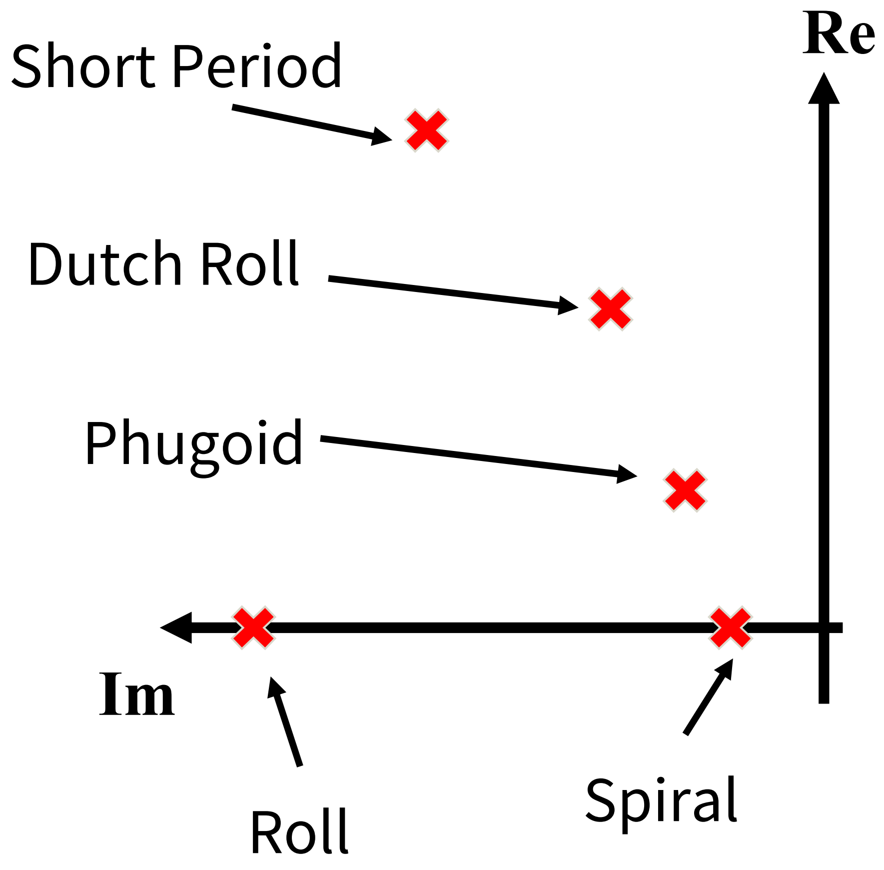
SUAVE

- **Objective:** Minimize C_D
- **Variables:** Wing twist $\theta_{1,2,..N}$; Span, b; Area, S; Aspect Ratio ; Taper $\boldsymbol{\lambda}$; Sweep γ; center of gravity location
- **Constraints:** $C_L = C_L$ @ cruise ; $C_M = 0$; $C_{M\alpha} < 0$; Static Margin = (c.g. - neutral point/MAC) > 10 %

Additional dynamic stability constraints: If we have confidence in moment of inertia, we can add additional constraints below. Taken from MIL-F-8785C for:

- Class I (Light utility)
- Category B (Non terminal flight phases) aircraft
- Level 1 flight quality (Flying qualities clearly adequate for the mission Flight Phase)

Additional Dynamic Stability Constraints



- $\lambda = n \pm i\omega$
- $T = 2\pi/\omega$
- $t_{double} = 0.693/|n|$
- $\omega_n = \text{sqrt}(\omega^2 + n^2)$
- $\zeta = -n/\omega$

Constraints:

- Damping ratio of phugoid mode (ζ_{ph}) > 0.04
- Frequency of short period mode(ω_{sp}) $\cong$ 1.34 rad/s
- Frequency of dutch roll (ω_d) > 1
- Time to double for spiral stability (T_{spiral}) > 4 sec
- Spiral stability criteria > 1

Elevator Optimization Setup

- Objective: Minimize elevator area
- Variables: Elevator span fraction start location, elevator span fraction end location, elevator chord fraction
- Constraints:
  - $\delta_{\text{e push}} < \delta_{\text{max}}$
  - $\delta_{\text{e pull}} < \delta_{\text{max}}$
  - *May want to include hinge moment constraint*

Elevator Optimization Setup

SUAVE

Where n_z are the load factors [3.0, -1]:

- 3.0 = stick pull (between 2.8 and 3.5)
- -1 = stick push (no more than -1)
- $C_{L\ maneuver}$ = computed in stability analysis

$$C_{L_{maneuver}} = n_z \frac{mg}{qS}$$

Aileron and Rudder Sizing: Part 1 (Roll)

SUAVE

- The user had the option of specifying presence of rudder. If rudder not presence, the optimization is done without this control surface
- Objective: Minimize total aileron (and rudder) control surface area
- Variables: Aileron span fraction start and end locations; aileron chord fraction; rudder span fraction start and end locations; rudder chord fraction
- Constraints:
  - $\delta_{\text{a roll}} < \delta_{\text{max}}$
  - $\frac{pb}{2V} > 0.07$ (this value is hard coded in stability analysis and the deflection is computed), where p = roll rate b = span ; V = cruise speed

Aileron and Rudder Sizing: Part 2 (Crosswind)

- Objective: Minimize total aileron (and rudder) control surface area
- Variables: Aileron span fraction start and end locations; aileron chord fraction; rudder span fraction start and end locations; rudder chord fraction
- Constraints: Trimmed with crosswind of 20 knots (input as sideslip angle into stability an analysis
  - δ_a crosswind < δ_{max}
  - δ_r crosswind < δ_{max}

Flap Sizing

SUAVE

- Objective: Minimize flap control surface area
- Variables: Flap span fraction start and end locations; flap chord fraction
- Constraints:
  - $\delta_{\text{flap}} < \delta_{\max}$
  - $\Delta C_{L_{max-flap}} \approx 0.7$ where $\Delta C_{L_{max-flap}} = 1.05\left(C_{L_{max-flap}} - C_{L_{max-no-flap}}\right)$

Optimization Pseudo-Algorithm Routine

- Starting with vehicle without control surfaces. This can be defined by user or read in through VSP. Optional assumptions include:
  - Wings are appropriately sized
  - Mass moment of inertia is defined
- Assign generic control surfaces to wings and keep track of:
  - Which control surfaces are appended to which wing
  - The function of the control surface (elevator for controlling pitching moment etc.)
- Optimize aircraft planform for stick-fixed stability and aerodynamics
  - Size elevator
  - Size aileron and rudder
  - Size Flap

SUAVE

File Structure in Repository

Optimize.py

Main script which packages objective function, constraints, design variables for solver

Mission.py

Defines the vehicle mission specifications

Vehicles.py

Reads in VSP aircraft and appends propulsion network for analysis

Procedure.py

Modifies vehicle each iteration, performance point analyses in some instances, stores updated geometry and post process results for solver

Control Surface Sizing Results : Twin-Engine CTOL CASE

- V_{cruise} = 120 mph
- n_z = [2.8, -1] (elevator) ; 2.8 (aileron & rudder)
- V_{max} = 1.4* V_{cruise}
- δ_{max}= $30°$ for ailerons, rudders & elevators, $40°$ for flap
- Crosswind = 20 kts (β =)

Optimization Summary

Optimization	**Time (min)**
Stick-fixed stability and aerodynamics	
Elevator	
Aileron & Rudder	
Flap	

Comparison of Results to Navion's Control Surface

SUAVE

	Optimization			True Values		
Control Surface	% Span Start Location	% Span End Location	Chord Fraction	% Span Start Location	% Span End Location	Chord Fraction
Elevator						
Aileron						
Rudder						
Flap						

3.9 SUAVE Tutorials

EVTOL ANALYSIS